

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-255066
(43)Date of publication of application : 01.10.1996

(51)Int.CI. G06F 3/14

(21)Application number : 07-350678 (71)Applicant : MICROSOFT CORP
(22)Date of filing : 13.12.1995 (72)Inventor : ORAN DANIEL P
IAN M ELLISON TAYLOR
CHI H CHU
JOSEPH D BELFIORE
MALAMUD MARK A
BOGDAN JEFFREY L
SULLIVAN KENT D

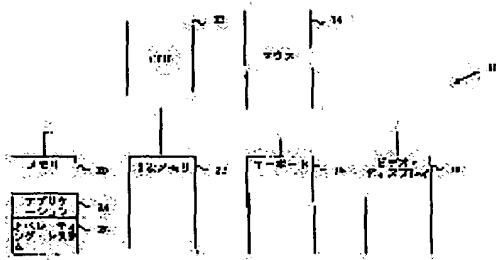
(30)Priority
Priority number : 94 354916 Priority date : 13.12.1994 Priority country : US

(54) TASK BAR WITH START MENU

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a task bar and a start menu provided with buttons corresponding to respective active tasks by using a computer system provided with an input device and a video display.

SOLUTION: The computer system is provided with a CPU 12 provided with access to some peripheral devices such as a mouse 14, a keyboard 16 and the video display 18. The mouse 14 is provided with at least left and right respective mouse buttons. This task bar is executed as a window by an operating system 26 and the window is generated by the operating system 26 at all times based on a window class. The task bar is an always visible user interface element, reports which task is active at present to a user and is not hidden by an opened window unless the user selects the hiding of the task bar.



LEGAL STATUS

[Date of request for examination] 02.09.1997
[Date of sending the examiner's decision of rejection] 02.12.2002

[Kind of final disposal of application other than
the examiner's decision of rejection or
application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of 2003-03393
rejection]
[Date of requesting appeal against examiner's decision of rejection] 03.03.2003
[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

フ

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-255066

(43)公開日 平成8年(1996)10月1日

(51)Int.Cl.⁶
G 06 F 3/14識別記号 340
府内整理番号F I
G 06 F 3/14技術表示箇所
340 B

審査請求 未請求 請求項の数42 O L 外国語出願 (全121頁)

(21)出願番号 特願平7-350678

(22)出願日 平成7年(1995)12月13日

(31)優先権主張番号 08/354916

(32)優先日 1994年12月13日

(33)優先権主張国 米国(US)

(71)出願人 391055933

マイクロソフト コーポレイション
MICROSOFT CORPORATIONアメリカ合衆国 ワシントン州 98052-
6399 レッドモンド ワン マイクロソフ
ト ウエイ (番地なし)

(72)発明者 ダニエル ピー オーラン

アメリカ合衆国 マサチューセッツ州
02139ケンブリッジ 406 フランクリン
ストリート 129

(74)代理人 弁理士 中村 稔 (外6名)

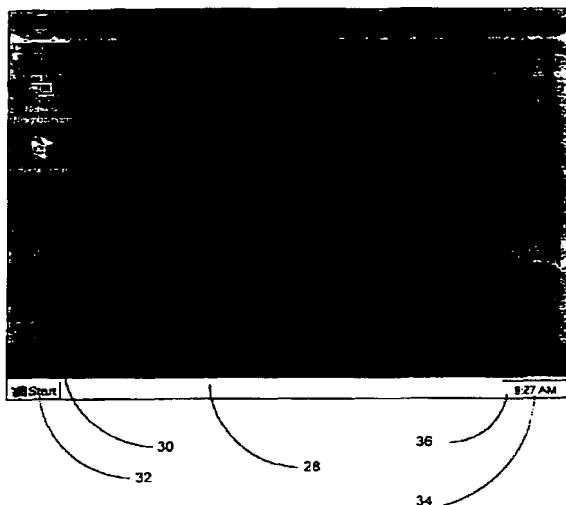
最終頁に続く

(54)【発明の名称】 スタート・メニュー付タスクバー

(57)【要約】

【課題】 アクティブ・ウィンドウを有する各アプリケーション用、ボタンのような視覚的な手掛けりをもたらすためにタスクバーを提供する。

【解決手段】 これらの視覚的な手掛けりは、ユーザに対して現在アクティブになっているウィンドウについての情報を与えるとともに、当該アクティブ・ウィンドウを表示するために使用することもできる。タスクバーには、プログラムのランチ、ドキュメントのオープンおよび、システム・セッティングのコントロールのためのメニューに対するユーザのアクセスを可能にするスタート・メニュー・ボタンを備えてもよい。このタスクバーは、サイズ変更が可能であり、ビデオ・ディスプレイ上の異なる位置にアンカーすることもできる。タスクバーは、他のウィンドウによる隠蔽が不可能なモードで動作可能である。さらに、他のウィンドウが非最小化状態にあるときは、このタスクバーを表示しないモードで動作させてもよい。



1

【特許請求の範囲】

【請求項1】 入力デバイスおよびビデオ・ディスプレイを有するコンピュータ・システムにおいて、ユーザ・インターフェース・エレメントをビデオ・ディスプレイ上に出力するステップであって、該ユーザ・インターフェース・エレメントは、該ビデオ・ディスプレイにおいて、ウインドウによる該ユーザ・インターフェース・エレメントの隠蔽が不可能なモードに置かれ、かつ該ユーザ・インターフェース・エレメントは、ユーザに対して現在アクティブ・ウインドウを有するアプリケーションを示すべく、このコンピュータ・システム内でアクティブ・ウインドウを有するアプリケーション用のサブエレメントを有するものとする、ユーザ・インターフェース・エレメントを該ビデオ・ディスプレイ上に出力するステップおよび、
ユーザによる該入力デバイスの使用に応じた、前記アプリケーションから選択した選択アプリケーション用の該サブエレメントの操作によって、該選択アプリケーションに、該選択アプリケーションの非最小化ウインドウの表示を要求するステップからなる方法。

【請求項2】 前記ユーザ・インターフェース・エレメントがウインドウとする、前記請求項1記載の方法。

【請求項3】 前記コンピュータ・システムが、少なくとも1つのマウス・ボタンを有するマウスを備え、該マウスは前記ビデオ・ディスプレイ上のカーソル操作用であり、前記アプリケーションから選択した選択アプリケーションに対する要求を行うステップが、該マウスのカーソルで該選択アプリケーションにポイントすべく該マウスのカーソルを位置決めするステップ、および、該マウスのボタンをクリックするステップよりなるものとする前記請求項1記載の方法。

【請求項4】 前記選択アプリケーションに対する要求を行うステップが、さらに、該選択アプリケーション用の前記サブエレメントの外観を変化して、該選択アプリケーション用のウインドウが可視であり、かつ非最小化されていることを示すステップを含むものとする、前記請求項1記載の方法。

【請求項5】 前記ユーザ・インターフェース・エレメントを出力するステップが、前記サブエレメントをボタンとして出力するステップを含むものとする、前記請求項1記載の方法。

【請求項6】 前記ユーザ・インターフェース・エレメントをビデオ・ディスプレイ上に出力するステップが、さらに、メニューにアクセスするためのサブエレメントを出力するステップを含み、該メニューが、ユーザによるシステム上のプログラムへのアクセスならびにシステムへのプログラムのランチを可能にするものとする、前記請求項1記載の方法。

【請求項7】 前記メニューが、最近開いたドキュメントのリストにアクセスするためのオプションを含むもの

とする、前記請求項6記載の方法。

【請求項8】 前記コンピュータ・システムが、さらに、入力デバイスを備え、前記方法が、ユーザによる該入力デバイスの使用に応じて、前記ユーザ・インターフェース・エレメントをリサイズするステップを含むものとする、前記請求項1記載の方法。

【請求項9】 前記入力デバイスがマウスであり、前記ユーザ・インターフェース・エレメントをリサイズするステップが、ユーザによるマウスを用いた該ユーザ・インターフェース・エレメントの境界のドラッグ・アンド・ドロップに応じて該ユーザ・インターフェース・エレメントをリサイズするステップよりなるものとする、前記請求項8記載の方法。

【請求項10】 前記ユーザ・インターフェース・エレメントをリサイズするステップが、該ユーザ・インターフェース・エレメントのサイズを増加するステップよりなるものとする、前記請求項8記載の方法。

【請求項11】 前記ユーザ・インターフェース・エレメントをリサイズするステップが、該ユーザ・インターフェース・エレメントのサイズを減少するステップよりなるものとする、前記請求項8記載の方法。

【請求項12】 前記ユーザ・インターフェース・エレメントをビデオ・ディスプレイ上に出力するステップが、さらに、ビデオ・ディスプレイ上のユーザ・インターフェース・エレメントの一部として、現在時を出力するステップを含むものとする、前記請求項1記載の方法。

【請求項13】 前記ユーザ・インターフェース・エレメントをビデオ・ディスプレイ上に出力するステップが、さらに、ユーザに対する視覚的な通知を行うために、ユーザ・インターフェース・エレメントの一部としてタスクバー通知エリアを前記ビデオ・ディスプレイ上に出力するステップを含むものとする、前記請求項1記載の方法。

【請求項14】 さらに、前記ユーザ・インターフェース・エレメントのモードを、ウインドウによる該ユーザ・インターフェース・エレメントの隠蔽が可能な第2のモードに変更するステップを含むものとする、前記請求項1記載の方法。

【請求項15】 さらに、前記ユーザ・インターフェース・エレメントのモードを、ウインドウが非最小化状態に置かれているとき、前記ユーザ・インターフェース・エレメントを可視としない第3のモードに変更するステップを含むものとする、前記請求項1記載の方法。

【請求項16】 さらに、ユーザによる前記入力デバイスの使用に応答して、前記ユーザ・インターフェース・エレメントを再表示するステップを含むものとする、前記請求項15記載の方法。

【請求項17】 入力デバイスおよびビデオ・ディスプレイを有するコンピュータ・システムにおいて、

10

20

30

40

50

3

該コンピュータ・システム上で実行するためのアプリケーションを提供するステップ

タスクバー・ユーザ・インターフェース・エレメントを該ビデオ・ディスプレイ上に出力するステップであって、該ユーザ・インターフェース・エレメントが、ウィンドウによるその隠蔽が不可能なモードに置かれるものとするタスクバー・ユーザ・インターフェース・エレメントを該ビデオ・ディスプレイ上に出力するステップ

該タスクバー・ユーザ・インターフェース・エレメント上に、アクティブ・ウィンドウを有する各アプリケーションのための視覚的表示を表示するステップユーザによる該入力デバイスの使用に応じて、選択アプリケーションの非最小化ウィンドウを表示するステップ

すでに閉じたアプリケーションをランチするステップおよび、

前記タスクバー・ユーザ・インターフェース・エレメントにすでに閉じたアプリケーション用の視覚的表示を表示するステップからなる方法。

【請求項18】さらに、すでに閉じたアプリケーションのランチに応じて、前記タスクバー・ユーザ・インターフェース・エレメント上の各視覚的表示をリサイズするステップを含むものとする、前記請求項17記載の方法。

【請求項19】さらに、追加の、すでに閉じたアプリケーションを開くステップおよび、

前記タスクバー・ユーザ・インターフェース・エレメント上に、該追加の、すでに閉じたアプリケーション用の視覚的表示を表示するステップを含むものとする、前記請求項17記載の方法。

【請求項20】前記視覚的表示がボタンである、前記請求項17記載の方法。

【請求項21】(a) ビデオ・ディスプレイ

(b) (i) アプリケーションおよび(ii) アクティブ・ウィンドウを有する各関連アプリケーション用のサブエレメントを表示するためのタスクバー・ユーザ・インターフェース・エレメントであって、ウィンドウによる隠蔽が不可能なモードに置かれるタスクバー・ユーザ・インターフェース・エレメントを表示するためのシステム・リソースを実行するプロセッサおよび、

(c) 前記サブエレメントの1つを操作し、該サブエレメントの関連アプリケーション用の非最小化ウィンドウを表示させるための入力デバイスを備えるコンピュータ・システム。

【請求項22】前記タスクバー・ユーザ・インターフェース・エレメントをウィンドウとする、前記請求項21記載のコンピュータ・システム。

【請求項23】前記サブエレメントをボタンとする、前記請求項21記載のコンピュータ・システム。

【請求項24】前記システム・リソースが、さらに、前記サブエレメントの関連アプリケーションのウインド

4

ウが非最小化されているとき、該サブエレメントの外観を変更する手段を含むものとする、前記請求項21記載のコンピュータ・システム。

【請求項25】前記システム・リソースは、さらに、新しいアプリケーションのランチがあったとき、該新しいアプリケーション用のサブエレメントを前記タスクバー・ユーザ・インターフェース・エレメントに追加する手段を含むものとする、前記請求項21記載のコンピュータ・システム。

【請求項26】前記システム・リソースが、さらに、選択アプリケーションのウィンドウが閉じているとき、該選択アプリケーション用のサブエレメントを削除する手段を含むものとする、前記請求項21記載のコンピュータ・システム。

【請求項27】前記システム・リソースが、さらに、前記タスクバー・ユーザ・インターフェース・エレメントをリサイズする手段を含むものとする、前記請求項21記載のコンピュータ・システム。

【請求項28】さらに、ユーザからの入力を受け取るための入力デバイスを備え、前記タスクバー・ユーザ・インターフェース・エレメントをリサイズする手段が、該入力デバイスよりのユーザ入力に応答して該タスクバー・ユーザ・インターフェース・エレメントをリサイズするものとする、前記請求項27記載のコンピュータ・システム。

【請求項29】前記システム・リソースが、さらに、前記タスクバー・ユーザ・インターフェース・エレメント上に現在時刻を示すクロックを表示するための手段を含むものとする、前記請求項21記載のコンピュータ・システム。

【請求項30】前記システム・リソースが、さらに、ユーザにプログラムへのアクセスを提供するスタート・メニューに対してアクセスするためのスタート・メニュー・エレメントを前記タスクバー・ユーザ・インターフェース・エレメント上に表示する手段を含むものとする、前記請求項21記載のコンピュータ・システム。

【請求項31】前記入力デバイスが、ビデオ・ディスプレイ上のカーソルを操作するための、マウス・ボタンを有するマウスによりなり、前記システム・リソースが、選択サブエレメントに対する該カーソルのポイント、およびユーザによる該マウス・ボタンのクリックに応答して、該選択サブエレメントの関連アプリケーション用の非最小化ウィンドウを表示する手段を含むものとする、前記請求項21記載のコンピュータ・システム。

【請求項32】前記タスクバー・ユーザ・インターフェース・エレメントが、前記ビデオ・ディスプレイ上の任意位置にアンカーされ、前記コンピュータ・システムが、前記ビデオ・ディスプレイ上の前記タスクバー・ユーザ・インターフェース・エレメントがアンカーされる位置を変更する手段を含むものとする、前記請求項21

5

記載のコンピュータ・システム。

【請求項33】 ビデオ・ディスプレイおよび、ユーザ入力を受け取るための入力デバイスを備えるコンピュータ・システムにおいて、

ビデオ・ディスプレイ上の任意位置にアンカーされるタスクバー・ユーザ・インターフェース・エレメントであって、アクティブ・ウィンドウを有する各アプリケーションを同定し、アプリケーション用の非最小化ウィンドウを可視とするために使用することができる視覚要素を備えるタスクバー・ユーザ・インターフェース・エレメントを提供するステップ

ユーザによる前記入力デバイスの使用に応じて、該タスクバー・ユーザ・インターフェース・エレメントを前記ビデオ・ディスプレイ上の他の位置にアンカーすべく、該タスクバー・ユーザ・インターフェース・エレメントを移動する手段を提供するステップおよび、

該タスクバー・ユーザ・インターフェース・エレメントを前記ビデオ・ディスプレイ上の新しい位置にアンカーすべく該タスクバー・ユーザ・インターフェース・エレメントを移動する手段を使用するステップからなる方法。

【請求項34】 ユーザ入を受け取るための入力デバイス、ビデオ出力を表示するためのビデオ・ディスプレイおよび、オペレーティング・システムを実行するためのプロセッサを備えるコンピュータ・システムにおいて、

該オペレーティング・システムにより、該ビデオ・ディスプレイ上に、スタート・メニューにアクセスするための視覚要素、および現在のアクティブ・タスクの表示を含むタスクバー・ユーザ・インターフェース・エレメントを表示するステップ

該スタート・メニューがドキュメント・オプションを含み、ユーザによる該スタート・メニューにアクセスするための該視覚要素の操作に対応する前記入力デバイスの使用に応じて、該スタート・メニューを表示するステップおよび、

ユーザによる前記入力デバイスを使用した前記ドキュメント・オプションの選択に応答して、最近開いたドキュメントのリストであって、それを介してユーザが該ドキュメントにアクセスし得るリストを表示するステップからなる方法。

【請求項35】 ユーザ入を受け付けるための入力デバイス、ビデオ出力を表示するためのビデオ・ディスプレイおよび、オペレーティング・システムを実行するためのプロセッサを備えるコンピュータ・システムにおいて、

該オペレーティング・システムにより、スタート・メニューにアクセスするための該視覚要素および現在のアクティブ・タスクの表示を含むタスクバー・ユーザ・インターフェース・エレメントを表示するステップ

6

ユーザによる該スタート・メニューへのアイテムの追加を可能にする該スタート・メニューを構成する手段を提供するステップ

ユーザの要求に対し、該スタート・メニューを構成する手段を使用して、該スタート・メニューに新しいアイテムを追加するステップおよび、

該新しいアイテムが追加された該スタート・メニューを表示するステップよりなり、該スタート・メニューは、ユーザによるプログラムおよびドキュメントへのアクセスのためのアイテムをも含むものとする方法。

【請求項36】 ユーザ入を受け取るための入力デバイスおよび、ビデオ出力を表示するためのビデオ・ディスプレイを備えるコンピュータ・システムにおいて、該ビデオ・ディスプレイ上に、現在のアクティブ・タスクの視覚的表示を有するタスクバー・ユーザ・インターフェース・エレメントを表示するステップ

ウィンドウを互いに関連させてどのように表示するかということを決定するためのアイテムおよび、該ウィンドウを互いに関連させてどのように表示するかということ

20 を決定するためのアイテムのうちの最後に選択したもの取消しを行うための選択アイテムを備えるコンテキスト・メニューを該タスクバー・ユーザ・インターフェース・エレメントに表示する手段を提供するステップ
該コンテキスト・メニューを表示する手段を使用して該コンテキスト・メニューを表示するステップおよび、ユーザによる前記選択アイテムの選択に応答して、前記ウィンドウを互いに関連させてどのように表示するかということを決定するためのアイテムのうちの最後に選択したもの効果の取消しを行うステップからなる方法。

【請求項37】 ユーザ入を受け取るためのマウスおよび、ビデオ出力を表示するためのビデオ・ディスプレイを備えるコンピュータ・システムにおいて、該ビデオ・ディスプレイ上にオブジェクトを表示するステップ

該ビデオ・ディスプレイ上に、アクティブ・ウィンドウを有する各アプリケーション用の視覚的表示を有するタスクバー・ユーザ・インターフェース・エレメントを表示するステップ

40 ユーザによる該マウスの使用に応じて該オブジェクトを該視覚的表示のうちの選択した選択視覚的表示に重なるべくドラッグし、ゼロでないしきい時間にわたり該選択視覚的表示に重なる位置を維持するステップおよび、該しきい時間にわたる該オブジェクトの該選択視覚的表示に重なる位置決めに応答して、該選択視覚的表示のアプリケーション用のウィンドウを開くステップからなる方法。

【請求項38】 さらに、ユーザによる前記マウスの使用に応じて、前記オブジェクトを前記選択視覚的表示のアプリケーション用のウィンドウ内にドロップするステップを含むものとする、前記請求項37記載の方法。

7

【請求項39】 入力デバイスおよびビデオ・ディスプレイを備えるコンピュータ・システムにおいて、

該ビデオ・ディスプレイ上に非最小化状態で表示されているアプリケーション・プログラム用のウィンドウがないとき、アクティブ・ウィンドウを有する各アプリケーション用の視覚的表示であって、関連づけられたアプリケーションを同定するため、および、該関連づけられたアプリケーションのウィンドウを可視の非最小化状態に置くための視覚的表示を備えるユーザ・インターフェース・エレメントを、該ビデオ・ディスプレイ上に表示するステップおよび、

いずれかのアプリケーションのウィンドウが可視の非最小化状態になると、該ビデオ・ディスプレイ上の該ユーザ・インターフェース・エレメントを隠すステップからなる方法。

【請求項40】 さらに、ユーザによる前記入力デバイスの使用に応じて、前記アプリケーションのいずれかが非最小化状態に置かれると、前記ユーザ・インターフェース・エレメントを再表示するステップを含むものとする、前記請求項39記載の方法。

【請求項41】 ユーザにより、前記ビデオ・ディスプレイ上のあらかじめ定義のあるエリアにカーソルが位置決めされると、前記ユーザ・インターフェース・エレメントを再表示するものとする、前記請求項39記載の方法。

【請求項42】 さらに、前記カーソルが前記ビデオ・ディスプレイ上のあらかじめ定義のあるエリア外となると、前記ユーザ・インターフェース・エレメントを再度隠すステップを含むものとする、前記請求項41記載の方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、データ・プロセッシング・システムに関し、特に、データ・プロセッシング・システムにおけるスタート・ウィンドウを備えるタスクバーの使用に関する。

【0002】

【従来の技術】 米国ワシントン州レッドモンドを居所とするMicrosoft Corporation (マイクロソフト・コーポレーション) から発売されている「MICROSOFT」ウィンドウズ3.1オペレーティング・システムは、ウィンドウ・ベースのグラフィカル・ユーザ・インターフェースをユーザに提供する。このオペレーティング・システムにより提供される環境においては、アプリケーション・プログラムが、ユーザがアプリケーションを使用するために非最小化フォームで「可視」とすることができる関連ウィンドウを備える。一般に、1つのアプリケーションは、1つの関連アイコンを有する。ユーザは、アプリケーションに対応するアイコンにマウスのカーソルを重ね、マウス・ボタンをダ

10

ブル・クリックすることによって当該アプリケーションを非最小化フォームで可視にすることができる。

【0003】

【発明が解決しようとする課題】 しかしながら、残念なことに、すべてのユーザがこのダブル・クリックを得手としているわけではない。たとえば、システムがダブル・クリックとして認識できるスピードでマウス・ボタンを2回クリックできないユーザがいる。また、ユーザがダブル・クリックの間にマウスを不用意に動かしたために、その動作がシステムに適正に伝わらないこともある。これとは別に、「MICROSOFT」ウィンドウズ3.1オペレーティング・システムのグラフィカル・ユーザ・インターフェースにおいて見られる困難性に、可視の非最小化ウィンドウおよび最小化ウィンドウ(つまりアイコン)が、他の非最小化ウィンドウもしくはアイコンによって隠蔽されてしまうことがある。この結果、不慣れなユーザには、いずれのアプリケーションが現在アクティブになっているかが分かりにくいことが少なくない。さらに、多くの初心者は、あるウィンドウが他を隠蔽していることに困惑する。これらのユーザは、隠蔽されているウィンドウに何が起こっているかわからない。また、多くの初心者は、この環境によって提供される階層に当惑している。この階層のトップは、使用可能な各プログラム・グループ用のアイコンを表示するプログラム・マネージャである。当該階層における次のレベルは、プログラム・グループによって形成されている。ユーザは、プログラム・グループ内のプログラムにアクセスする場合、通常、当該プログラム・グループに関連づけられているウィンドウを開く。プログラム・グループのウィンドウには、当該プログラム・グループ内で使用可能な各アプリケーションに対応するアイコンが表示される。この後ユーザは、ファイル・メニューを介して、あるいはプログラム・アイコンをダブル・クリックすることにより、そのうちのいずれかのプログラムを開くことになる。ところで、この階層を横切ることは、多くのユーザにとって極めて困難であり、大抵は適正に通過する方法がわからなくなり、階層の中で迷ってしまう。

20

【0004】 この従来のアプローチに伴う他の複雑化の要因は、いかにして、あるいは、どこでヘルプ情報を得るべきか、ユーザにわからないことである。さらに、ドキュメントを開いたりプログラムを切り替える際にもユーザには困難が強いられている。

30

【0005】 【課題を解決するための手段】 本発明の第1の特徴は、入力デバイスおよびビデオ・ディスプレイを備えるコンピュータ・システムを使用する方法である。この方法においては、ユーザ・インターフェース・エレメントが、ビデオ・ディスプレイ上の出力となる。このユーザ・インターフェース・エレメントは、ウィンドウが隠蔽する

40

50

ことができないモードに置かれる。ユーザ・インターフェース・エレメントには、当該コンピュータ・システム内でアクティブ・ウィンドウを有する各アプリケーションに対応するサブエレメントが備わる。このサブエレメントは、ユーザに対して、現在アクティブ・ウィンドウを有しているアプリケーションを示す。選択したアプリケーションの非最小化ウィンドウを表示する要求は、ユーザによる入力デバイスの使用に応じた当該サブエレメントの操作により発生する。本発明の別の特徴による方法は、コンピュータ・システム上でアプリケーションの実行がある環境において実現される。ここでは、ユーザ・インターフェース・エレメント用のタスクが、ウィンドウが隠蔽することができないモードに置かれる。タスクバー・ユーザ・インターフェース・エレメント上には、アクティブ・ウィンドウを有しているアプリケーションのそれぞれに対し、ビジュアル・インジケータが表示される。選択したアプリケーション用の非最小化ウィンドウの表示は、ユーザによる入力デバイスの使用に応じてなされる。アプリケーションを開じると、その後タスクバー・ユーザ・インターフェース・エレメント上にランチされ、当該アプリケーションに対応するビジュアル・インジケータが表示される。

【0006】本発明のさらに別の特徴によれば、コンピュータ・システムがビデオ・ディスプレイおよびプロセッサを備える。このプロセッサは、アプリケーションおよびタスクバー・ユーザ・インターフェース・エレメント表示用のシステム・リソースの実行に使用される。タスクバー・ユーザ・インターフェース・エレメントは、アクティブ・ウィンドウを有している各関連アプリケーションに対応するサブエレメントを表示する。このタスクバー・ユーザ・インターフェース・エレメントは、ウィンドウが隠蔽することができないモードに置かれる。また、コンピュータ・システムは、サブエレメントの1つを操作し、当該サブエレメントの関連アプリケーション用の非最小化ウィンドウを表示するための入力デバイスも備えている。本発明の他の特徴によれば、ユーザ・インターフェース・エレメント用のタスクが、ビデオ・ディスプレイ上の任意の位置にアンカーされる。このタスクバー・ユーザ・インターフェース・エレメントには、アクティブ・ウィンドウを有している各アプリケーションを同定するビジュアル・エレメントが備わる。これらのビジュアル・エレメントは、当該アプリケーション用に可視の非最小化ウィンドウを生成するために使用することができる。この場合、タスクバー・ユーザ・インターフェース・エレメントをビデオ・ディスプレイ上の他の位置にアンカーできるように、ユーザによる入力デバイスの使用に応じてタスクバー・ユーザ・インターフェース・エレメントを移動する手段が備わる。この手段は、タスクバー・ユーザ・インターフェース・エレメントをビデオ・ディスプレイ上の新しい位置にアンカー

すべく移動される。

【0007】本発明のさらに別の特徴によれば、タスクバー・ユーザ・インターフェース・エレメントがオペレーティング・システムによってビデオ・ディスプレイ上に表示される。タスクバー・ユーザ・インターフェース・エレメントには、スタート・メニューにアクセスするためのビジュアル・エレメントおよび現在アクティブとなっているタスクのインジケータが備わる。スタート・メニューは、当該スタート・メニューへのアクセスのためのビジュアル・エレメントの操作に対応するユーザによる入力デバイスの使用に応じて表示される。このスタート・メニューには、ドキュメント・オプションが備わる。ユーザが入力デバイスを使用してこのドキュメント・オプションを選択すると、それに応答して、過去に開いたドキュメントのうち、時間的に最も新しいドキュメントのリストが表示され、ユーザはそれにアクセスすることができる。本発明のまた別の特徴によれば、スタート・メニューにアクセスするためのビジュアル・エレメントおよび現在アクティブとなっているタスクのインジケータを包含すべくタスク・ユーザ・インターフェースが表示される。当該スタート・メニューに新しいアイテムを追加するために、ユーザは、必要に応じてスタート・メニューを構成する手段を使用することができる。これによりスタート・メニューは、新アイテムを擁した状態で表示される。また、スタート・メニューは、ユーザによるプログラムおよびドキュメントへのアクセスを可能にするアイテムも備えている。

【0008】本発明の特徴の1つによれば、タスクバー・ユーザ・インターフェース・エレメントがビデオ・ディスプレイ上に表示される。このタスクバー・ユーザ・インターフェース・エレメントは、現在アクティブとなっているタスクのビジュアル・インジケータを備える。ここでは、当該タスクバー・ユーザ・インターフェース・エレメント用のコンテキスト・メニューを表示するための手段が使用される。このコンテキスト・メニューは、互いの相関から複数のウィンドウをいかに表示すべきかを決定するアイテムを備える。また、コンテキスト・メニューは、互いの相関から複数のウィンドウをいかに表示すべきかを決定するアイテムのうち、最後に選択したもののが取消を行なわれる。本発明の別の特徴によれば、オブジェクトがビデオ・ディスプレイ上に表示される。また、ビデオ・ディスプレイ上には、タスクバー・ユーザ・インターフェース・エレメントも表示される。このタスクバー・ユーザ・インターフェース・エレメントは、アクティブ・ウィンドウを有する各アプリケーションのためのビジュアル・インジケータを備える。ユーザは、

11

マウスを使用してビデオ・ディスプレイ上に表示されるオブジェクトをドラッグし、タスクバー・ユーザ・インターフェース・エレメント上のビジュアル・インジケータのうちの選択したものの上に重ねる。この際、当該オブジェクトを、選択したビジュアル・インジケータ上に所定しきい時間を超えて重ねておく。オブジェクトを、所定しきい時間を超えて選択したビジュアル・インジケータ上に重ねて置くと、それに応答して当該ビジュアル・インジケータのアプリケーション用のウィンドウが開かれる。

【0009】本発明のさらに別の特徴によれば、現在、非最小化状態で可視となっているアプリケーション・プログラム用のウィンドウがまったくない場合に、ユーザ・インターフェース・エレメントがビデオ・ディスプレイ上に表示される。このユーザ・インターフェース・エレメントには、関連アプリケーションを同定するためのアクティブ・ウィンドウを有する各アプリケーション用のビジュアル・インジケータ、および関連アプリケーションのウィンドウを開くためのビジュアル・インジケータが備わる。ビデオ・ディスプレイ上のユーザ・インターフェース・エレメントの表示は、これらのアプリケーションのうちの1つに対応するウィンドウが開かれると隠される。

【0010】

【実施例】本発明の好ましい実施例は、従来のシステムで問題となっていた困難性の多くを克服するタスクバーおよびスタート・メニューを提供する。このタスクバーは、終始可視となるユーザ・インターフェース・エレメントであり、ユーザに対して現在いずれのタスクがアクティブになっているかを知らせる。タスクバーは、ユーザがタスクバーの隠蔽を選択しない限り、開いているウィンドウによって隠蔽されることがないように構成されている。またタスクバーは、現在のアクティブ・タスク・ウィンドウを表示するためのポイントとしても機能する。タスクバーは、各アクティブ・タスクに対応するボタンを備える。タスク・ウィンドウを表示する場合、ユーザは、当該タスクに与えられているタスクバー・ボタンをシングル・クリックするだけでよい。これらのボタンは、タスクの切り替えに使用することもできる。タスクバーは、ユーザ・インターフェース上の固定位置にアンカーされた状態になっている。しかしながら、ユーザは、この固定アンカ位置を変更することができる。さらにタスクバーには、スタート・メニュー・ボタンが備わり、これは、ユーザ・インターフェースを介してプログラムにアクセスするための集中ロケーションとして働く。スタート・メニューは、ユーザによる所望のプログラム、ドキュメントもしくはシステムコントロールに対する容易なアクセスを可能にするメニュー・アイテムを備える。このスタート・メニューは、詳細を後述するように、構成を変更することができるようになっている。

12

【0011】図1は、本発明の好ましい実施例を実行するのに適したコンピュータ・システムを示すブロック図である。このコンピュータ・システムは、マウス14、キーボード16およびビデオ・ディスプレイ18を始めとするいくつかの周辺装置へのアクセスを有する中央処理装置(CPU)12を備える。マウス14は、少なくとも左マウス・ボタンおよび右マウス・ボタンを備えるものとする。またCPU12は、メモリ20および2次メモリ22へのアクセスも有している。メモリ20は、10 アプリケーション24およびオペレーティング・システム26のコピーを保持する。本発明の好ましい実施例においては、タスクバーおよびスタート・メニューがオペレーティング・システム26のシェルによって供給される。しかしながら、当業者であれば本発明がオペレーティング・システムにおける実施に限定されることは容易に理解されよう。むしろ本発明は、ユーティリティないし他のシステム・リソースを介して実現される可能性もある。なお以下に述べる説明においては、このオペレーティング・システムを、Microsoft Corporation(マイクロソフト・コーポレーション)が提供する「MICROSOFT」ウィンドウズ95オペレーティング・システムとする。ただし、当業者においては、本発明を他のオペレーティング・システムによっても実施し得ることを理解されたい。

【0012】タスクバーは、オペレーティング・システム26によりウィンドウとして実行される。ウィンドウは、ウィンドウ・クラスに基づき、オペレーティング・システム26によって常に生成される。ウィンドウ・クラスは、当該ウィンドウに送られるメッセージを処理するウィンドウ・プロシージャを同定する。オペレーティング・システム26は、アプリケーションによるウィンドウ・クラスの記録を可能にするRegisterClass()関数をサポートしている。ウィンドウ・プロシージャは、ウィンドウに送られるメッセージの受け取りおよび処理を行う関数である。オペレーティング・システム26は、メッセージ・ベースのオペレーティング・システムであり、そのオペレーションは、ウィンドウ間でのメッセージのバスにより達成される。図2は、本発明の好ましい実施例のオペレーティング・システム26によって提供されるグラフィカル・ユーザ・インターフェースを一例で示したものである。タスクバー28は、ビデオ・ディスプレイ18(図1)上のパーキャル・デスクトップ30の下端に位置決めされている。図2に示した例においては、ユーザ・インターフェースの左端から右端までタスクバー28が延びており、ビデオ・ディスプレイの下端にアンカーされている。またタスクバー28は、全長にわたり均一な幅を有している。タスクバー28は、スタート・メニューへアクセスするためのスタート・メニュー・ボタン32を備える。このスタート・メニューについては、詳細を後述する。さらにタ

スクバー28には、クロック34およびタスクバー通知エリア36が備わる。このクロック34は、現在時刻を表示する。また、タスクバー通知エリア36には、イベントの可視表示として機能するアイコン、すなわちステータス情報を伝えるアイコンが表示される。タスクバー通知エリアについては、本件特許出願人と共通の出願人により同時出願なされた「System Provided Visual Notification Area (システム供給ビジュアル表示エリア)」と題したシリアル番号 の出願明細書に詳説されている。なお、スタート・メニュー・ボタン32およびクロック34は、タスクバー28が所定の位置にアンカーされている限り、当該タスクバー内の固定位置に保持される。

【0013】タスクバー・ボタンおよびスタート・メニュー・ボタン32は、本件特許出願人と共通の出願人により同時出願なされた「System Provided Child Window Controls (システム供給チャイルド・ウィンドウ・コントロール)」と題したシリアル番号 の出願明細書に詳説されているようなタブ・コントロールとして実行される。コントロールは、ペアレン特・ウィンドウ内に存在するチャイルド・ウィンドウであり、当該チャイルド・ウィンドウ内でユーザからの入力等のイベントが発生した場合に、通知メッセージをペアレン特・ウィンドウに送る。具体的には、チャイルド・ウィンドウは、マウスならびにキーボードのメッセージを処理し、チャイルド・ウィンドウの状態に変化があったとき、ペアレン特・ウィンドウに通知する。これにより、チャイルド・ウィンドウをペアレン特・ウィンドウの入力デバイスの一種として動作させることができになる。前述したようにタスクバー28は、システム内における現在のアクティブ・タスクをすべて同定するための集中ロケーションとして機能する。図3は、ワードプロセッサ・プログラムおよび電子メール・プログラムがアクティブになっている場合のタスクバー28の外観の一例である。タスクバー28は、図3に示したボタン38およびボタン40のようなタスクバー・ボタンの表示によって、アクティブとなっているタスクを示す。ボタン38ならびに40の境界は、3次元的な効果をもたらすとともに、タスクのウィンドウが開いていないことが示されるように、1段高くなつた状態に描かれている。ユーザは、マウス14もしくは他の適当な入力デバイスを使用してマウスのカーソル41でタスクバー・ボタンをポイントすることによりタスクバー28からアクティブ・タスク・ウィンドウを表示し、その後、左マウス・ボタンをシングル・クリックして当該タスクに関連づけられているウィンドウを開き、可視の非最小化状態での表示を行う。たとえば、ワードプロセッサ・プログラムのみがアクティブになっており、ユーザが当該ワードプロセッサ・プログラム用のボタン38にマウスのカーソル41を重ねて左マウス・ボ

タンをシングル・クリックした場合、図4に示すように、当該ワードプロセッサ・プログラム用のウィンドウ42が開かれる。さらに、ワードプロセッサ・プログラム用のタスクバー・ボタン38は、そのボタンが押し込まれたようにリドローされる。この結果、ユーザは、当該タスクに関連づけられているウィンドウが現在アクティブであることを示す視覚的な手掛かりを得ることができる。

【0014】プログラミングの見地からは、タスクバー上にボタンを有するタスクに対応するウィンドウの表示において、以下に示すステップが実行される。まず、マウスのカーソルがタスクバー・ボタンをポイントすべく位置決めされると、チャイルド・ウィンドウ・コントロールとしてタスクバー・ボタンがマウス・メッセージを受け取る。続いて、タスクバー・ボタンは、左マウス・ボタンがクリックされたことを示すマウス・メッセージを受け取る。このマウス・メッセージは、タスクバー28のマネージャーと表示を担当するペアレン特・タスクバーの処理に送られる。タスクバー28は、当該マウス・メッセージを受け取ると、それに応答して図4に示したようなワードプロセッサ・ウィンドウを開くための判断を行う。さらに、ボタンが押し込まれた状態が示されるように、チャイルド・ウィンドウ・コントロールの外観を変更する。タスクバー・ボタンのサイズは、タスクバー28上で使用可能なスペースに応じて変化する。図3では、タスクバー・ボタン38および40が最大許容サイズで示されている。しかしながら、さらに別のタスクがアクティブになった場合は、それらのタスク用のタスクバー・ボタンがタスクバー28上に追加されなければならない。多くの場合、すべてのアクティブ・タスクを適切に表示するために、新規のサイズでのタスクバー・ボタンのリドローが必要になる。たとえば、図5に、前述のプログラムに加えて、スケジュール・プログラムおよび計算機プログラムがアクティブになった場合の例を示す。つまり、新規にアクティブとなったこれらのタスクに対応するタスクバー・ボタンもタスクバー28上に表示されなければならない。そこで、すべてのアクティブ・タスクに対応するすべてのタスクバー・ボタンを適切に表示するために、すべてのタスクバー・ボタンのサイズが変更される。つまり、図5に示したタスクバー・ボタン38、40、44および46の幅は、図3に示したタスクバー・ボタン38および40の幅より狭いものとなる。

【0015】タスクバー・ボタンのリサイズが必要な場合の新規サイズを決定するために、タスクバーはタスクバー28内のタスクバー・ボタン表示用の使用可能スペースを算出する。続いてタスクバーは、表示の必要なタスクバー・ボタンの数を判断し、使用可能スペースを表示すべきタスクバー・ボタンの数で除する。この後タスクバーは、この演算に基づいてタスクバー・ボタンの適

15

切なサイズを決定する。ビデオ・ディスプレイ18の下端にタスクバー28がアンカーされてタスクバー・ボタンが表示されている場合、左揃えの態様でボタンが表示される。オペレーティング・システム26は、チャイルド・ウィンドウ・コントロールとして各タスクバー・ボタンをリドローするメカニズムを提供する。基本的にタスクバー・ボタンのリドローは、タスクバー・ボタンに関連づけられている各ウィンドウのリサイズを伴う。ユーザは、タスクバー28の内側境界を移動してタスクバーのサイズを増減することができる。この場合の内側境界とは、グラフィカル・ユーザー・インターフェースの残りの部分に面している境界のことである。図7に示した例においては、タスクバー28の上側の境界が内側境界54となる。図6に示したステップを実行することにより、ユーザは、タスクバーの境界を断続的に増加（つまり、図7に示す最低高によるステップ増加）してその位置を再設定することができる。最初にユーザは、カーソルを移動してタスクバーの内側境界をポイントする（ステップ48）。これにより、従来型の矢印カーソル（図3に示したカーソル41参照）が2重の矢印に変わり、タスクバーの内側境界を移動するポジションであることが示される。この後ユーザが、左マウス・ボタンを押し下げたままタスクバーが所望のサイズとなるまでドラッグし（ステップ50）、左マウス・ボタンをリリースしてタスクバーの内側境界をドロップすると、タスクバーのリサイズが行なわれる（ステップ52）。図7に示したように、このドラッグ・アンド・ドロップ・オペレーションの間は、境界のゴースト56が表示されて、2重矢印カーソル58の現在位置によって示される新規の内側境界の位置の表示が行なわれる。

【0016】前述したように内側境界は、2重矢印カーソル58の現在位置に基づいて決定される断続的な位置にのみ移動させることができる。また内側境界を、タスクバーが多くともグラフィカル・ユーザ・インターフェースの半分を占める位置を超えて移動させることはできない。図8は、一例として、これまでの図面で示したタスクバーの高さの2倍となるように内側境界をドラッグした後のタスクバー28を示したものである。タスクバー28がウィンドウであることから、タスクバーのリサイズは、他のウィンドウのリサイズと同様に扱われる。タスクバー28のマネージおよび表示を担当するオペレーティング・システム26におけるタスクバー処理は、ドラッグ・アンド・ドロップ・オペレーションおよびタスクバー・ウィンドウのリサイズに係るマウス・メッセージを受け取る。前述のようにタスクバーは、ビデオ・ディスプレイ18の各端にアンカーすることができる。タスクバー28を別の辺にアンカーするための移動を行う場合、ユーザは、図9に示すようなステップを実行する。具体的には、カーソルをタスクバー28の位置にポイントさせる。続いて左マウス・ボタンを押し下げ、そ

10

のままタスクバーのアンカーを希望する辺までタスクバーをドラッグする（図9のステップ62）。そこでタスクバーをドロップすれば、タスクバーがリドローされて適切なエッジにアンカーされる（ステップ64）。システムは、スクリーンを論理的に4つの象限に分割している。各象限は、各辺に対応している。タスクバーは、ドロップの時点でマウス・カーソルのホット・スポットがある象限に対応する辺にアンカーされる。図10は、図8に示したタスクバーをビデオ・ディスプレイ18の右端にドラッグしたときの外観を示している。ここで、タスクバーがアンカーされている位置に応じてボタン38、40、44および66のリドローが行なわれ、位置が再設定される点に注意されたい。タスクバー・ボタンは、アンカーが行なわれる各辺に対して水平に延びるように表示される。スタート・メニュー・ボタン34およびクロックならびにタスクバー通知エリアもまた、それに応じて位置の再設定が行なわれる。図11は、タスクバー28をビデオ・ディスプレイ18の上端にドラッグしたときの外観の一例を、図12は、タスクバー28をビデオ・ディスプレイ18の左端にドラッグしたときの外観の一例をそれぞれ示している。

【0017】ユーザがマウスのカーソルを移動させてタスクバー・ボタンの1つをポイントし、右マウスボタンを押し下げるとき、コンテキスト・メニューが表示される。図13に一例として、マウスのカーソルでタスクバー・ボタン44をポイントし、右マウスボタンを押し下げた場合に表示されるコンテキスト・メニュー66を示す。ユーザは、このコンテキスト・メニュー66によってコンテキストで区別されたオペレーション（当該タスクバー・ボタンに関連づけられたウィンドウ上で実行することができる）の選択が可能になる。たとえば、コンテキスト・メニュー66には、タスクバー・ボタン44に関連づけられたウィンドウを閉じるオプション「閉じる」が備わる。タスクバー28は、コンテキスト・メニュー68（図14）も備える。このタスクバー28は、ユーザがマウスのカーソルでタスクバーをポイントし右マウスボタンを押し下げたときに表示される。コンテキスト・メニュー68は、開いているウィンドウを階層表示する「階層」オプションを備えている。また、タスクバー用のコンテキスト・メニュー68は、開いているウィンドウを水平にタイリングする「水平に並べる」オプションも備える。さらに、開いているウィンドウを垂直にタイリングする「垂直に並べる」オプションも備える。これらのオペレーションは、従来のテクニックを使用して実行される。「全ウィンドウの最小化」オプションは、現在開いているすべてのウィンドウを最小化し、タスクバー28上のタスクバー・ボタンとして表示するオプションである。オペレーティング・システム26は、ユーザが使用できる定義可能作業エリアという概念を包含している。ウィンドウは、作業エリア内に留まる

20

20

30

40

50

16

ことのないように最小化される。つまり、最小化されたウィンドウは、ユーザから不可視となる。またタスクバーには最小化要求が知られ、タスクバーは、対応するボタンが一段高く見えるように表示すべく各ウィンドウ用のタスクバー・ボタンをアップデートする。

【0018】コンテキスト・メニュー68には、図15に示すように、直前に選択したウィンドウのタイリング・オプションを取り消すための「タイリングの取消し」オプションを備えることもできる。たとえば、図15に示した例においては、カレンダー・プログラム用のウィンドウ70および計算機プログラム用のウィンドウ72が開かれている。この例は、「垂直に並べる」オプションが選択された後の状態である。これらのウィンドウは、コンテキスト・メニュー68にある「タイリングの取消し」オプションを選択することによって、垂直タイリングの要求を発する前の状態に表示し直すことができる。システムは、直前に選択されたウィンドウのタイリング・オプションを記録し、取消し要求があった後に記録しているタイリング・オプションを再実行することによって、このタイリングの取消しの効果を達成している。タスクバーのコンテキスト・メニュー68における最後のオプションは、「プロパティ」オプションであり、これによりタスクバー28およびスタート・メニュー用のプロパティ・シートを表示させることができる。図16に、タスクバーもしくはコンテキスト・メニュー68から「プロパティ」オプションを選択した場合に現れる表示を示す。ここには、「タスクバーのプロパティ」というタイトルの付いたフレーム・ウィンドウ74が表示されている。フレーム・ウィンドウ74は、フレームのようにしてプロパティ・シートを保持するために使用されている。「タスクバー・オプション」用のプロパティ・シート76は、「スタート・メニュー・プログラム」プロパティ・シート用のタブ78とともにフレーム・ウィンドウ74内に表示される。このウィンドウには、ユーザが行ったプロパティの変更を受け入れるOKボタン84、ユーザが行ったプロパティの変更をキャンセルするキャンセル・ボタン82および、フレーム・ウィンドウ74を開いた状態で変更を直ちに適用するための適用ボタン84が備わっている。

【0019】「タスクバー・オプション」プロパティ・シート76は、プレビュー・エリア86を備えており、ここにはチェック・ボックス88、90および92の選択または非選択の効果が表示される。図17は、ユーザ・インターフェース内でタスクバーを他の要素の前面に表示する「常に前へ」チェック・ボックス88がチェックされた状態を一例で示している。プロパティ・シート76のプレビュー・エリア86は、そこに表示されているタスクバー94の一部がウィンドウ96を隠蔽する姿勢となることを示している。この態様は、図16に示した、タスクバーがウィンドウによって隠蔽されるプレビ

ュー・エリア86におけるタスクバーの態様に逆になっている。図18は、タスクバーをオート・ハイド・モードにする「自動で隠す」チェック・ボックス90を選択した場合の一例を示している。プロパティ・シート76のプレビュー・エリア86によって示されるように、オート・ハイド・モードにおいてはタスクバーの顕著な表示がなくなる。この場合、タスクバーの細長いバーだけが可視となる。このオート・ハイド・オプションによって、ユーザは、必要に応じてビデオ・ディスプレイ18の使用可能スクリーン・スペースを広げることが可能である。

【0020】オート・ハイド・モードでは、タスクバーが隠される。タスクバー28は、ユーザがデスクトップ上にウィンドウを開いたときに隠される。しかし、開いているウィンドウがなくなると、タスクバーが再び現れる。またユーザは、ウィンドウが開いていても、ビデオ・ディスプレイ上のタスクバーがアンカーされている辺をマウスのカーソルでポイントすることによってタスクバーを表示させることができる。この場合、ユーザがカーソルをこのエリアから外に出すとタスクバーは隠れる。プログラム的には、オート・ハイド・モードは、ウィンドウが開いている場所の監視および、バーを表示する時期ならびにタスクバーを表示する時期の決定により実行される。図19は、スタート・メニューにラージ・アイコンに代えてスマール・アイコンを表示させる「スタート・メニューにスマール・アイコンを表示」リスト・ボックス92を選択した例を示している。システムは、2種類のアイコン、すなわちスマール・アイコンおよびラージ・アイコンを備えている。スマール・アイコンは、その呼称からもわかるように、寸法においてラージ・アイコンと異なり、それよりも小さいアイコンとなっている。チェック・ボックス92をチェックすると、プロパティ・シート76のプレビュー・エリア86が、スタート・メニュー100内にスマール・アイコンを表示する。

【0021】スタート・メニュー・ボタン32は、スタート・メニューを開くための手段として機能する。スタート・メニューは、ユーザがプログラム、ドキュメント、システムのセッティング、ヘルプ情報等にアクセスするための中心的な場所として動作する。マウスのカーソルでスタート・メニュー・ボタン32をポイントし、左マウス・ボタンをクリックすると、スタート・メニュー101(図20)が表示される。スタート・メニュー101は、デフォルトで多数のメニュー・アイテムを備えている。ユーザは、「検索」メニュー・アイテム104を用いて、プログラム、ドキュメントもしくはその他のシステム内の構成要素を検索するための問い合わせダイアログを呼び出すことができる。また「ヘルプ」メニュー・アイテム106を用いれば、ユーザは、各種のトピックについての有用な情報が得られるメニュー・ダイ

19

アログを呼び出すことができる。つまり、ヘルプもしくは検索が必要な場合、ユーザは、スタート・メニュー12にアクセスし、これらのメニュー・アイテムの1つを選択するだけでよい。「設定」メニュー・アイテム108では、ユーザによるシステム・セッティングの調整が可能になる。「ドキュメント」メニュー110においては、最大15までの最近使用したドキュメントへのアクセスが可能になる。図21は、「ドキュメント」メニュー・アイテムの選択により、階層表示のメニュー103が3つのドキュメントを表示している例である。システム16は、最近使用した15のドキュメントおよびそれらのアイコンのキャッシュを保持している。このリストのドキュメント用のメニュー・アイテムは、「ドキュメント」メニュー・アイテム110が選択されたとき表示される。

【0022】「プログラム」メニュー・アイテム112では、スタート・メニュー101から階層表示されたプログラム・メニュー105(図22)へのアクセスが可能になる。プログラム・メニュー105は、ユーザの選択肢を与えるプログラムおよびプログラム・グループを表示する。ユーザは、「実行」メニュー・アイテム114によりプログラムを選択して実行することができる。最後に、「終了」メニュー・アイテム116は、終了ダイアログを表示し、ユーザはこれによりシステムを終了することができる。メニュー・アイテム104、108、110および112に対応して表示されている矢印118は、これらのメニュー・アイテムが階層メニューを備えていることを示しており、マウスのカーソルでこれらのメニュー・アイテムをポイントするとメニューが階層表示される。図23は、スタート・メニュー101内の「設定」メニュー・アイテム108に関するメニュー120の階層表示の例を示している。スタート・メニュー101内のメニュー・アイテムの1つをマウスのカーソルでポイントすると、そのアイテムはハイライト表示される。図23においては、「設定」メニュー・アイテム108がハイライト表示され、それに対応するメニュー120の階層表示により使用可能な設定オプションがユーザに示される。コントロール・パネル・ウィンドウを開くためのオプション122もこの設定オプションの中にある。これらのオプションは、マウスのカーソルでポイントすることによりハイライト表示される。図24に示した例においては、コントロール・パネル・オプション122がハイライト表示で示されている。この状態、すなわちコントロール・パネル・オプション122をマウスのカーソルでポイントしている状態でユーザが左マウスボタンをシングル・クリックすると、(図25に示したように)コントロール・パネル・ウィンドウ124が開かれる。

【0023】スタート・メニュー101内に現れる上述のメニュー・アイテムは、デフォルトで表示されるもの

10 (11)

20

である。ユーザは、アイテムの追加または削除により、スタート・メニューを変更することができる。ただし特定のアイテムについては、システムでプロテクトを設定し、ユーザによるスタート・メニュー101からのそれらのアイテムの削除を不可能にしている。ユーザは、スタート・メニューの構成を変更する場合、タスクバー・コンテキスト・メニュー68を介して「スタート・メニュー・プログラム」プロパティ・シートを呼び出す。

「スタート・メニュー・プログラム」プロパティ・シート128は、すでにタスクバー用のプロパティ・シートに関して説明したフレーム・ウィンドウ74の中に表示される。「スタート・メニュー・プログラム」プロパティ・シート128は、スタート・メニューにアイテムを追加するための「追加」ボタン130を備えている。またここには、スタート・メニューからアイテムを削除するための「削除」ボタン132および、スタート・メニューの高度なカスタム化を実行するための「詳細」ボタン134も備わる。ユーザによる「追加」ボタン130の押し下げがあると、図27に示したように、スタート20・メニューにアイテムを追加するためのシーケンスの一部であるダイアログ138が表示される。ダイアログ138は、追加すべきアイテムの名称を入力するための編集ボックス140を備えている。ここへの入力に代えて、ブラウズ・ボタン142をアクティブにすれば、ユーザによる追加すべきアイテムのブラウズを可能にする追加のダイアログが呼び出される。さらにダイアログ138には、当該シーケンス内の次のダイアログに進むための「次へ」ボタン146および、当該シーケンス内の直前のダイアログに戻るための「戻る」ボタン144が備わる。最後に、スタート・メニューへのアイテムの追加をキャンセルするための「キャンセル」ボタン148が備えられている。

【0024】ユーザがブラウズ・ボタン142(図27)をクリックすると、ブラウズ・ダイアログ150(図28)が表示される。ブラウズ・ダイアログ150は、スタート・メニュー101に追加することができるフォルダおよびアイテムを表示するウィンドウ154を備えている。ユーザは、ウィンドウ154から表示されているアイテムの1つを選択しても、また編集ボックス40152内にファイル名を入力してもよい。その後、「開く」ボタン156をクリックすれば、選択アイテムをスタート・メニュー101に追加する旨を要求することができる。また、キャンセル・ボタン158をクリックすれば、ブラウズ・ダイアログ150がキャンセルされる。ブラウズ・ダイアログ150には、ドロップ・ダウン・リスト・ボックス160が備わり、これにより、スタート・メニューにアイテムを追加するために参照する場所に関するオプションがユーザに与えられる。これとは別に、ドロップ・ダウン・リスト・ボックス162は、ユーザによるウィンドウ154に表示すべきファイ

21

ルの種類の選択のために備えられている。図29に、ユーザがスタート・メニューにアイテムを追加するため、「Windows」というフォルダを開き、「Calc」というプログラムを選択した場合の例を示す。具体的には、ユーザが「Calc」に関係づけられているアイコンをクリックしたことを示す。この結果、このファイルの名称が編集ボックス152に現れる。

【0025】ユーザがブラウズ・ダイアログ150内の「開く」ボタン156をクリックすると、「プログラム・グループの選択」ダイアログ160が表示される。このダイアログは、スタート・メニュー101からアクセス可能な階層表示のメニュー内の、アイテムを追加すべきプログラム・グループを選択するために使用する。図30に示した例においては、スクロール・ボックス168内に表示されたスタート・メニュー用のエントリにマウスのカーソルをポイントし、左マウス・ボタンをクリックすることによって、スタート・メニュー101をプログラム・グループとして選択した状態が示されている。この場合ユーザは、「新しいフォルダ」ボタン166をクリックして、階層表示のスタート・メニューに追加する新規のフォルダを定義してもよい。ユーザが「次へ」ボタン146をクリックすれば、選択アイテムがスタート・メニュー101に追加される。図26において、ユーザが「削除」ボタン130を選択した場合は、「ショートカット/フォルダの削除」ダイアログ170(図31)が表示される。このダイアログ170は、現在スタート・メニュー101からアクセスすることができるアイテムのツリー表示172を表示する。スタート・メニューからアイテムを削除する場合は、ツリー階層表示172からアイテムを選択し、「削除」ボタン174をクリックすればよい。選択しているアイテムは、そのラベルがハイライト表示される。図31に示した例においては、「Calc」プログラムが選択されている。ダイアログ170を終了する場合は、「閉じる」ボタン176を選択すればよい。

【0026】「詳細」ボタン134(図26)を選択した場合は、スタート・メニューを詳細に調べるためにウインドウ180(図32)が表示される。このエクスプローラ・ウインドウ180に現れるアイテムは、OLEおよびウインドウズのドラッグ・アンド・ドロップ・オペレーションをサポートしていることから、これらのアイテムのドラッグ・アンド・ドロップによりスタート・メニューを操作することができる。ユーザは、フレーム・ウインドウ74(図26)のプロパティ・シートの編集が完了したら、その変更を受け入れる場合は「OK」ボタン80を、キャンセルする場合は「キャンセル」ボタン82をクリックする。ここで、上記の例においてユーザがプロパティ・シートへの変更を通してスタート・メニューへの「Calc」プログラムの追加を決定したものとする。これによりスタート・メニュー101は、

22

図33に示すような態様となり、「Calc」プログラム用のメニュー・アイテム182が追加される。この後は、「Calc」プログラムにアクセスする場合、階層表示のメニューを縦横することなく、初期スタート・メニュー101から直接行うことができるようになる。

【0027】上記の例は、全体として、タスクバー28がユーザ・インターフェースの下端に位置決めされている場合について述べてきた。しかしながら、タスクバー28がアンカーされている位置に応じて、スタート・メニュー101も異なった位置に階層表示される点に注意されたい。図34に、タスクバー28がユーザ・インターフェースの右端にアンカーされている場合のスタート・メニュー101の例を示す。スタート・メニュー101は、オペレーティング・システム26のタスクバー処理のコントロール下にある。つまり、スタート・メニューをどこに表示するかということは、タスクバーが判定する。したがって、スタート・メニューを表示する位置は、タスクバー28を移動し、別の位置にアンカーすると変更される。なお、一般に、スタート・メニュー101の表示は、タスクバー28の内側となる。タスクバー28にあるクロック34は、関連づけられたプロパティ・シートを有する。これらのプロパティ・シートへのアクセスは、クロック34のダブル・クリックもしくはコンテキスト・メニュー188(図35)の呼び出しによって得られる。コンテキスト・メニュー188は、ユーザがマウスのカーソルでクロック34をポイントし、右マウス・ボタンを押し下げるとき表示される。このコンテキスト・メニュー188には、「日付/時刻の調整」メニュー・アイテム190が備わる。クロック34用のプロパティ・シートには、「日付および時刻」プロパティ・シート190(図36)および、「時間帯」プロパティ・シート192(図37)がある。ユーザは、これらのプロパティ・シートにより、現在の年、月、日、時、分、秒および時間帯を選択することができる。

【0028】前述したように、タスクバー28は、開いているウインドウによって安易に隠蔽されることなく、一貫して可視となるようにデザインされている。図38は、ワード・プロセッサ・プログラム用のウインドウが最大表示されている場合を示すが、この場合にもタスクバー28が、最大表示のワード・プロセッサ・プログラム用のウインドウ200により隠蔽されることはない。これは、アプリケーションが使用できる作業エリアからタスクバーの領域を除くことにより達成される。なお、タスクバーがワード・プロセッサ・プログラム用のウインドウを覆うこともなく、これらの境界は、単に接するだけである。タスクバーに関する起起される1つの複雑性は、ユーザが現在最小化されているアクティブ・ウインドウにあるオブジェクトをドラッグする場合に係る。本発明の好ましい実施例においては、ユーザがアプリケーションに関連づけられているタスクバー・ボタン

にオブジェクトをドラッグすることができる。具体的には、図39に示すように、まずマウス14を使用してオブジェクトをドラッグし、いずれかのタスクバー・ボタンに重ねる(図39、ステップ202)。この状態、すなわちマウス・ボタンを押し下げ、当該オブジェクトをタスクバー・ボタンに重ねた状態を、所定時間継続する(ステップ204)。この状態で所定時間が経過すると、当該タスクバー・ボタンに関連づけられているアプリケーション・ウィンドウが非最小化状態で可視となり(ステップ206)、ユーザは、このアプリケーション・ウィンドウ内に当該オブジェクトをドロップすることができる(ステップ208)。

【0029】以上は、好ましい実施例を参照した本発明についての説明である。当業者であれば、本件特許請求の範囲から逸脱することなく、実施態様や詳細における変形が可能であることが理解されよう。

【図面の簡単な説明】

【図1】本発明の好ましい実施例を実行するのに適したコンピュータ・システムを示すブロック図である。

【図2】本発明の好ましい実施例のタスクバーを採用したユーザ・インターフェースを一例で示す平面図である。

【図3】2つのアプリケーションが開かれ、それらのウィンドウが最小化されている場合の、図2に示したタスクバーの外観を示す平面図である。

【図4】プログラムがタスクバー上のタスクバー・ボタンから完全にユーザの管理下となる態様を一例で示す平面図である。

【図5】タスクバー上の使用可能スペースに基づいてタスクバー・ボタンが自動的にリサイズされる態様を一例で示す平面図である。

【図6】タスクバーをリサイズするとき実行されるステップを説明するためのフローチャートである。

【図7】タスクバーのリサイズが行なわれる場合のタスクバーの外観を一例で示す平面図である。

【図8】図7に示したタスクバーとは異なるサイズのタスクバーの外観を一例で示す平面図である。

【図9】タスクバーをビデオ・ディスプレイの新しいコーナーに再アンカーするとき実行されるステップを説明するためのフローチャートである。

【図10】ビデオ・ディスプレイの右端にアンカーされたタスクバーを示す平面図である。

【図11】ビデオ・ディスプレイの上端にアンカーされたタスクバーの外観を示す平面図である。

【図12】ビデオ・ディスプレイの左端にアンカーされたタスクバーの外観を示す平面図である。

【図13】タスクバー・ボタン用のコンテキスト・メニューを一例で示す平面図である。

【図14】タスクバー用のコンテキスト・メニューを一例で示す平面図である。

【図15】ウィンドウのタイリング・オプションが選択されたときのタスクバー用のコンテキスト・メニューを一例で示す平面図である。

【図16】タスクバー用の「タスクバー・オプション」プロパティ・シートを示す平面図である。

【図17】図14に示したプロパティ・シートにある「常に前へ」チェック・ボックスを選択した場合の効果を説明するための平面図である。

【図18】図14に示したプロパティ・シートにある「自動で隠す」チェック・ボックスを選択した場合の効果を説明するための平面図である。

【図19】図14に示したプロパティ・シートにある「スタート・メニューにスマート・アイコンを表示」チェック・ボックスを選択した場合の効果を説明するための平面図である。

【図20】本発明の好ましい実施例によるスタート・メニューを一例で示す平面図である。

【図21】スタート・メニューを介してアクセス可能な階層表示のドキュメント・メニューを一例で示す平面図である。

【図22】スタート・メニューを介してアクセス可能な階層表示のプログラム・メニューを一例で示す平面図である。

【図23】スタート・メニューを介してアクセス可能な階層表示の設定メニューを一例で示す平面図である。

【図24】図17に示した階層表示の設定メニューにおいて「コントロール・パネル」オプションが選択された場合を示す平面図である。

【図25】図18に示した階層表示の設定メニューを介してアクセス可能なコントロール・パネルが開かれた状態を示す平面図である。

【図26】スタート・メニュー用の「スタート・メニュー・プログラム」プロパティ・シートを示す平面図である。

【図27】図20に示したプロパティ・シートにある「追加」ボタンを介してアクセス可能なダイアログを示す平面図である。

【図28】スタート・メニューにアイテムを追加するためのブラウズ・ダイアログを示す平面図である。

【図29】図22に示したブラウズ・ダイアログをスタート・メニューに追加するアイテムの選択に使用する場合を示す平面図である。

【図30】スタート・メニューにアイテムを追加するための一連のダイアログにおける別のダイアログを示す平面図である。

【図31】スタート・メニューからアイテムを除去するためのダイアログを示す平面図である。

【図32】スタート・メニューの高度な操作を実行するためのオープン・エクスプローラ・ウィンドウを示す平面図である。

25

【図33】アイテムをスタート・メニューに追加すべく構成されたスタート・メニューを一例で示す平面図である。

【図34】タスクバーがビデオ・ディスプレイの右端にアンカーされている場合のスタート・メニューの階層表示を示す平面図である。

【図35】タスクバー上のクロックに対して生成可能なコンテキスト・メニューを示す平面図である。

【図36】「日付および時間」プロパティ・シートを一

26

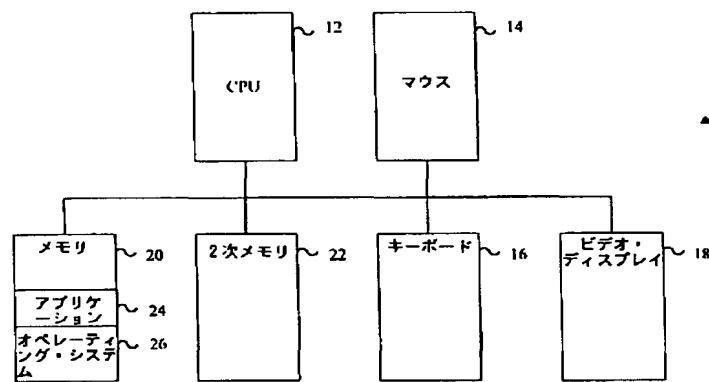
例で示す平面図である。

【図37】タスクバー上のクロック用の「時間帯」プロパティ・シートを示す平面図である。

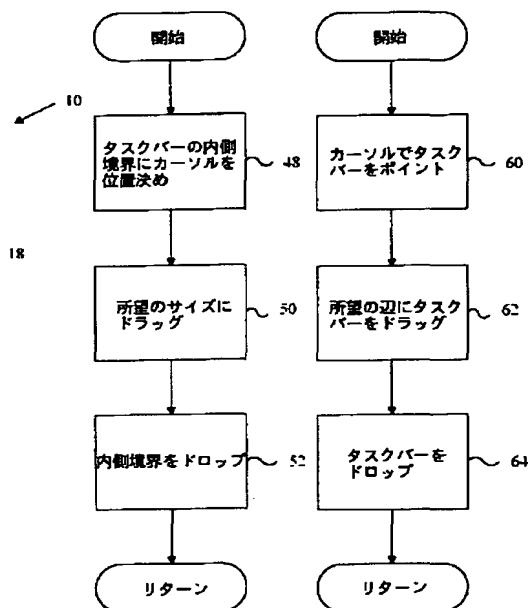
【図38】ワードプロセッサ・アプリケーションによりタスクバーが隠蔽されない方法を示す平面図である。

【図39】タスクバー・ボタンにオブジェクトをドラッグするために実行されるステップを説明するためのフローチャートである。

【図1】

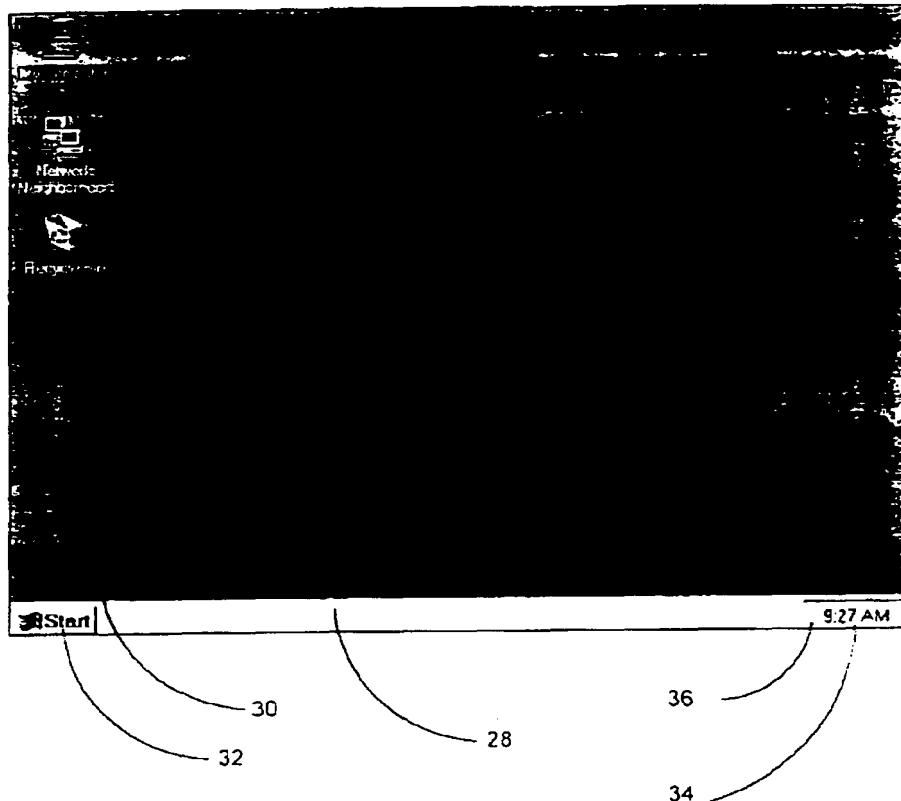


【図6】

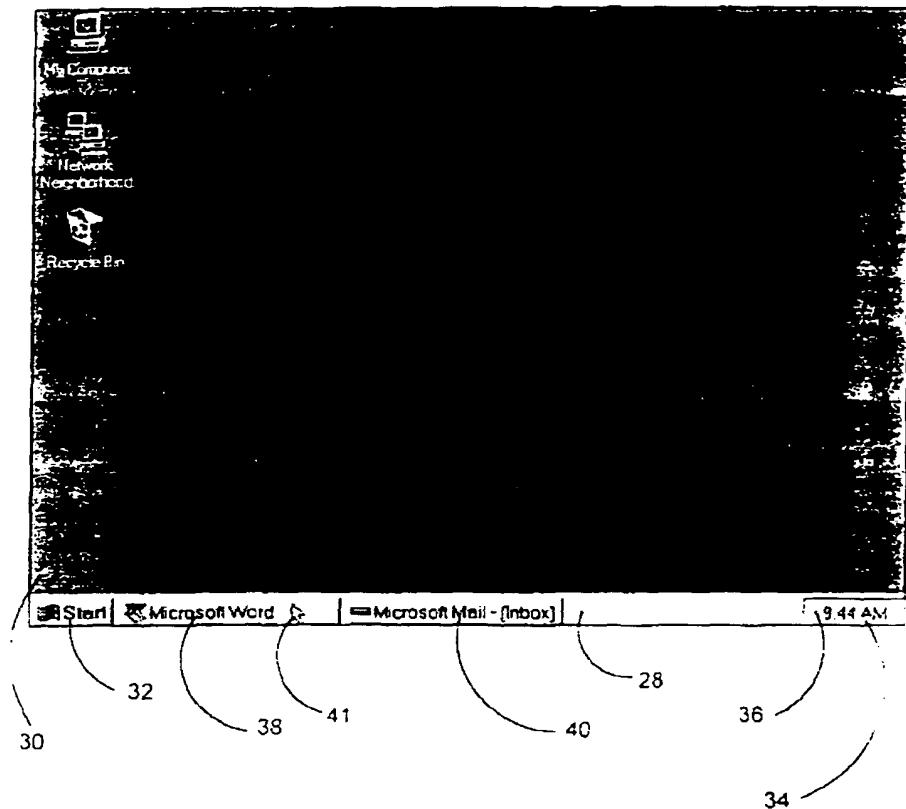


【図9】

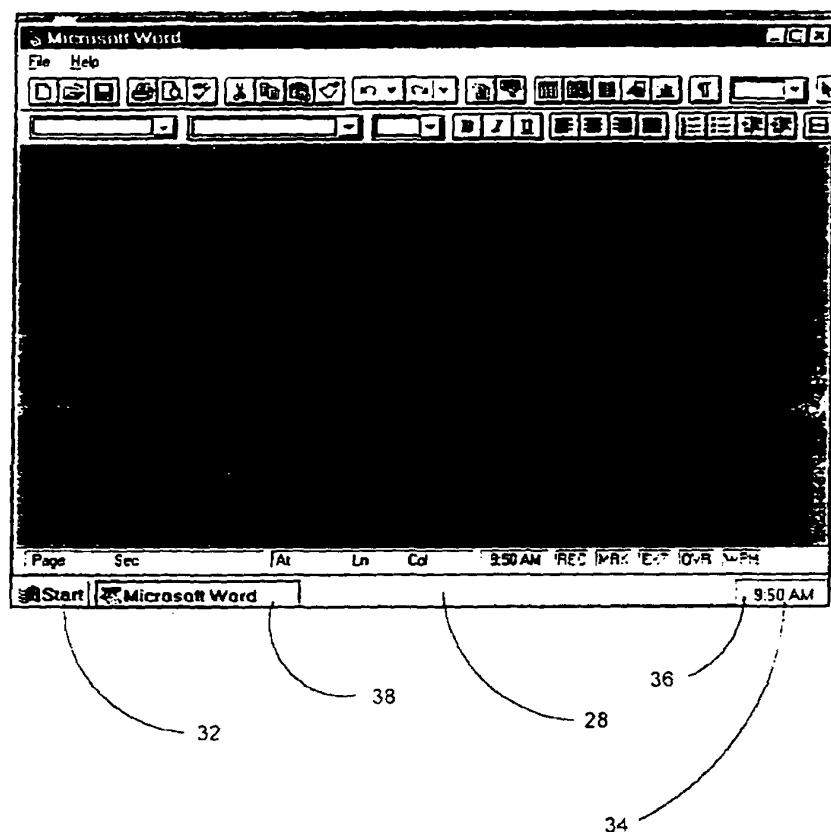
【図2】



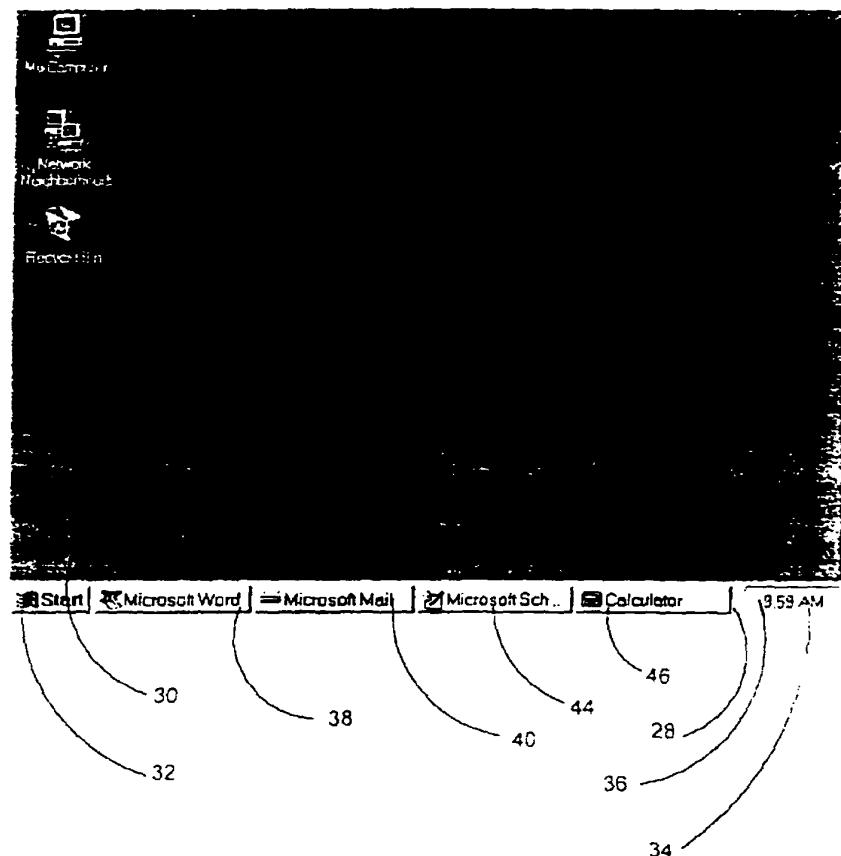
【図3】



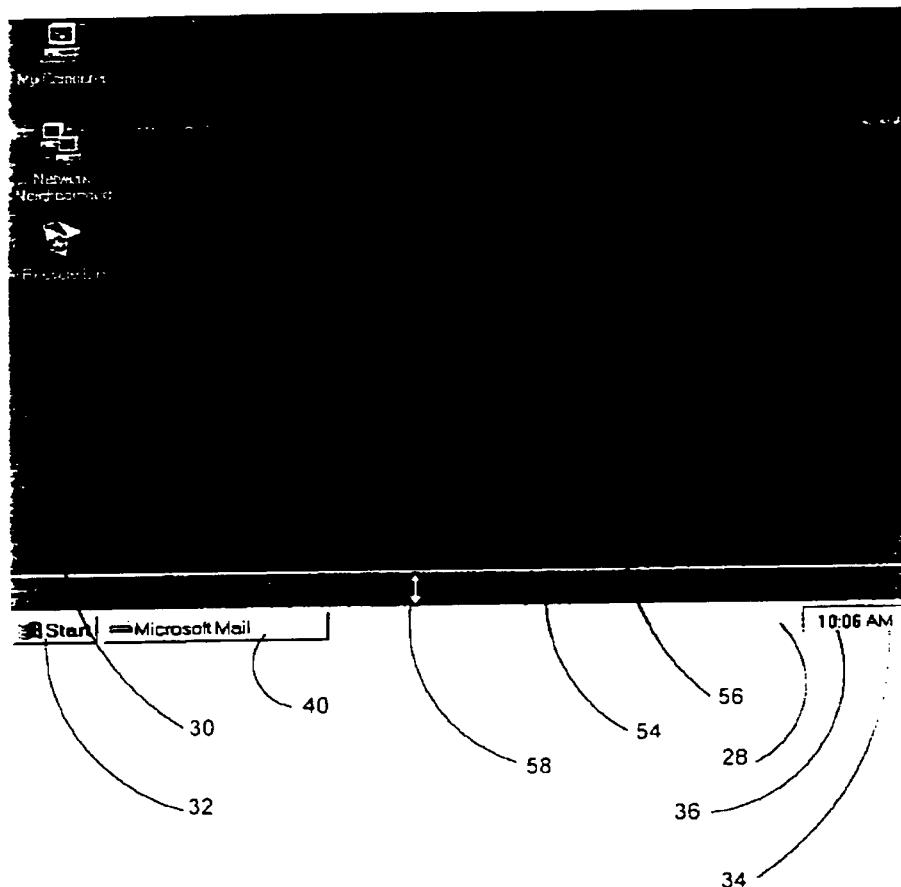
【図4】



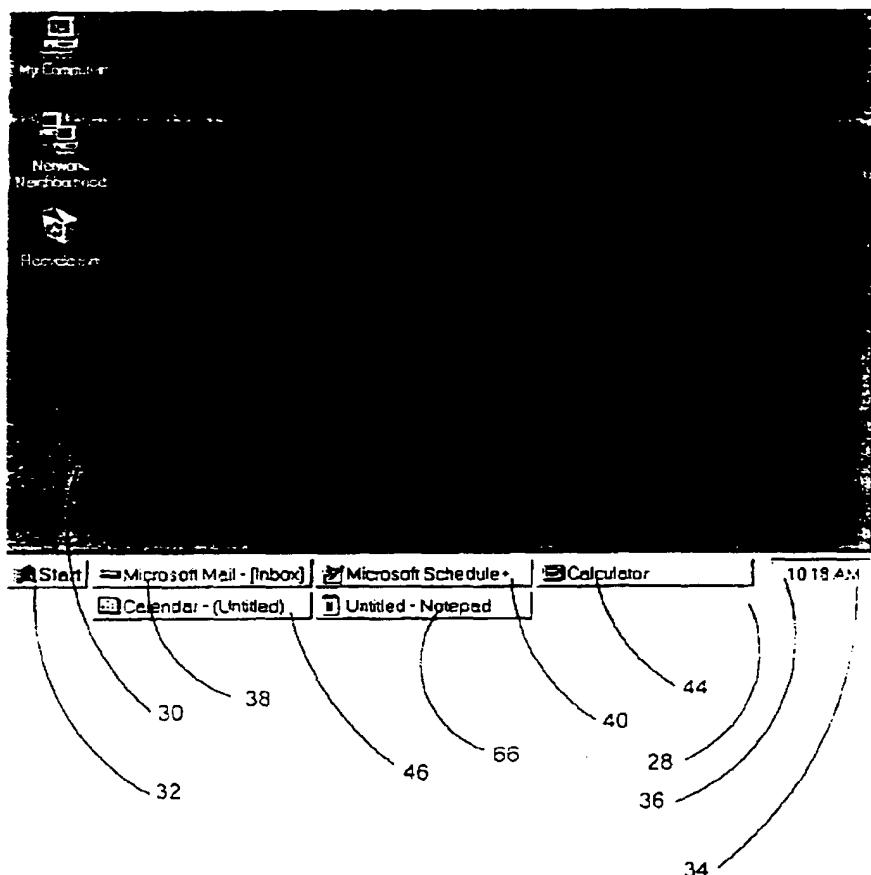
【図5】



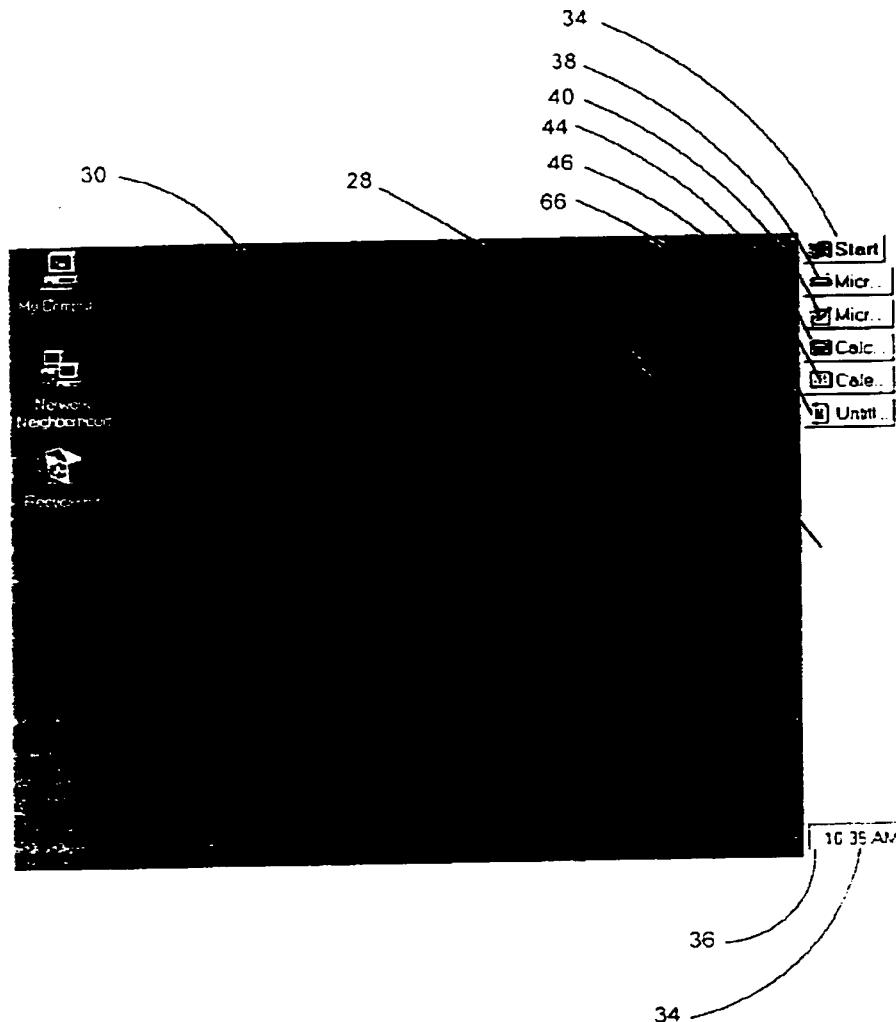
【図7】



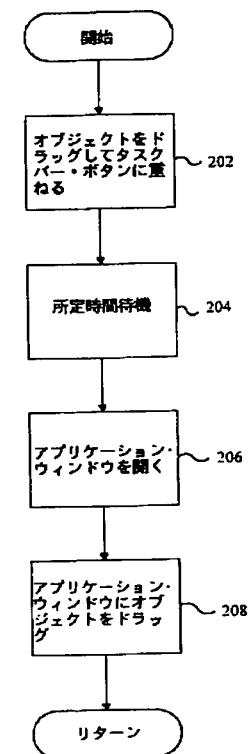
【図8】



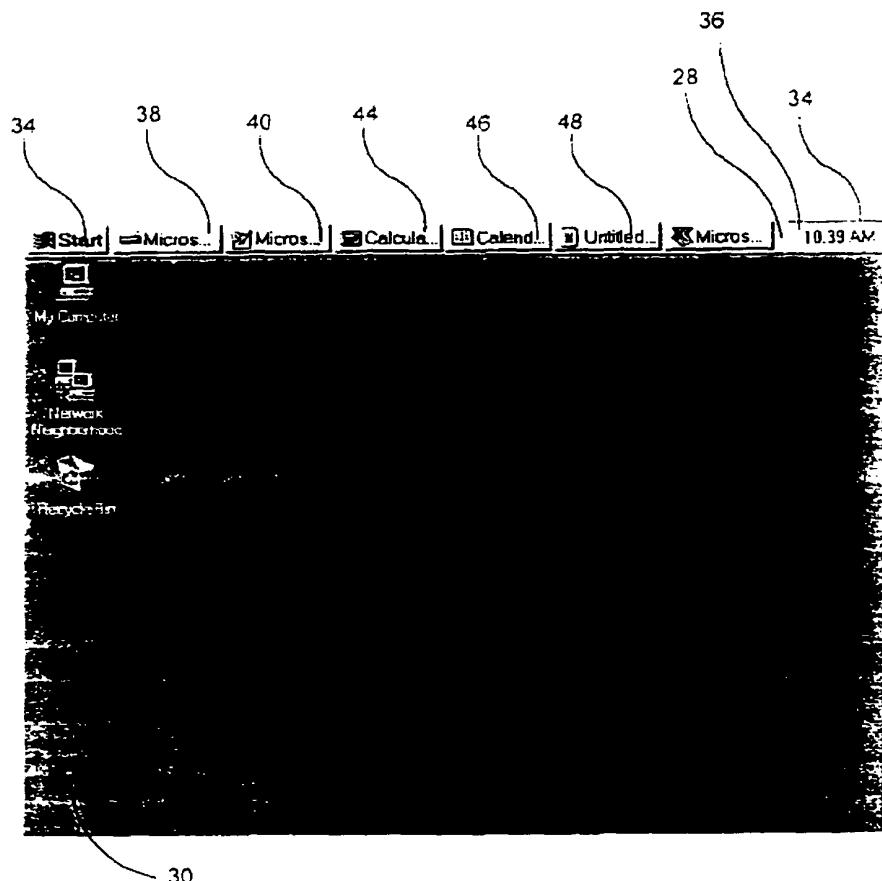
【図10】



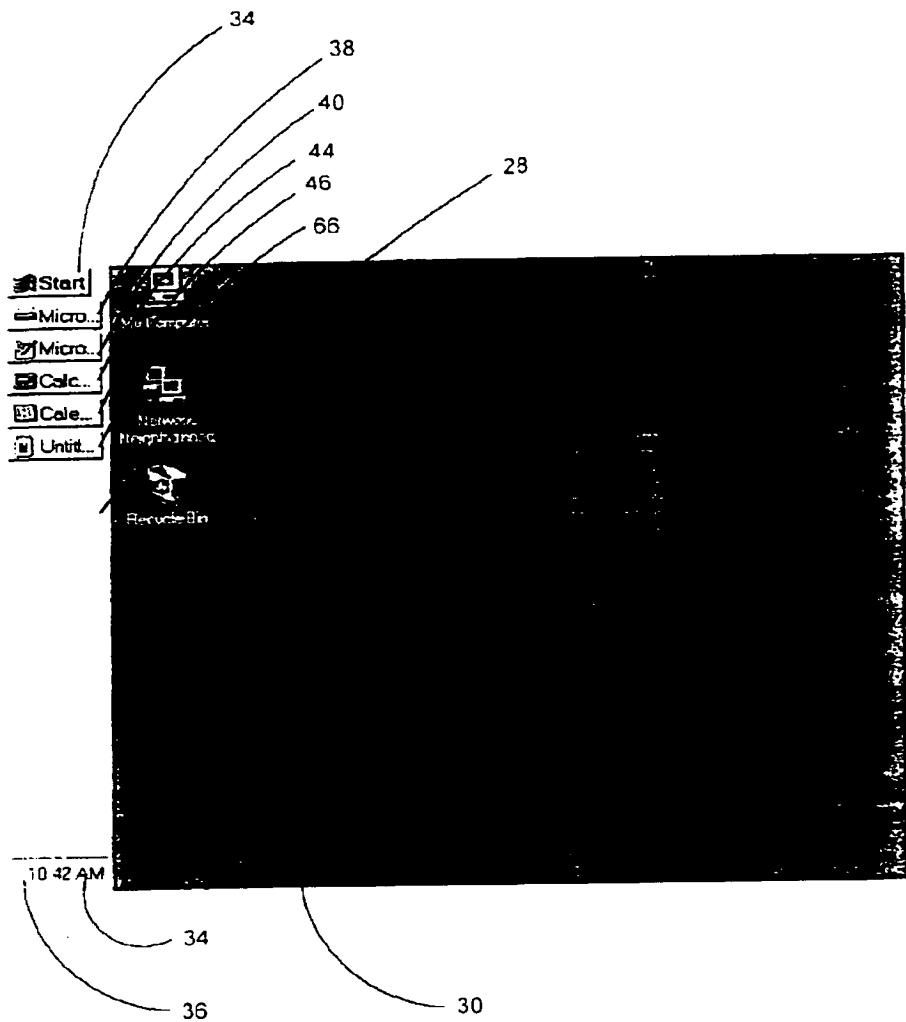
【図39】



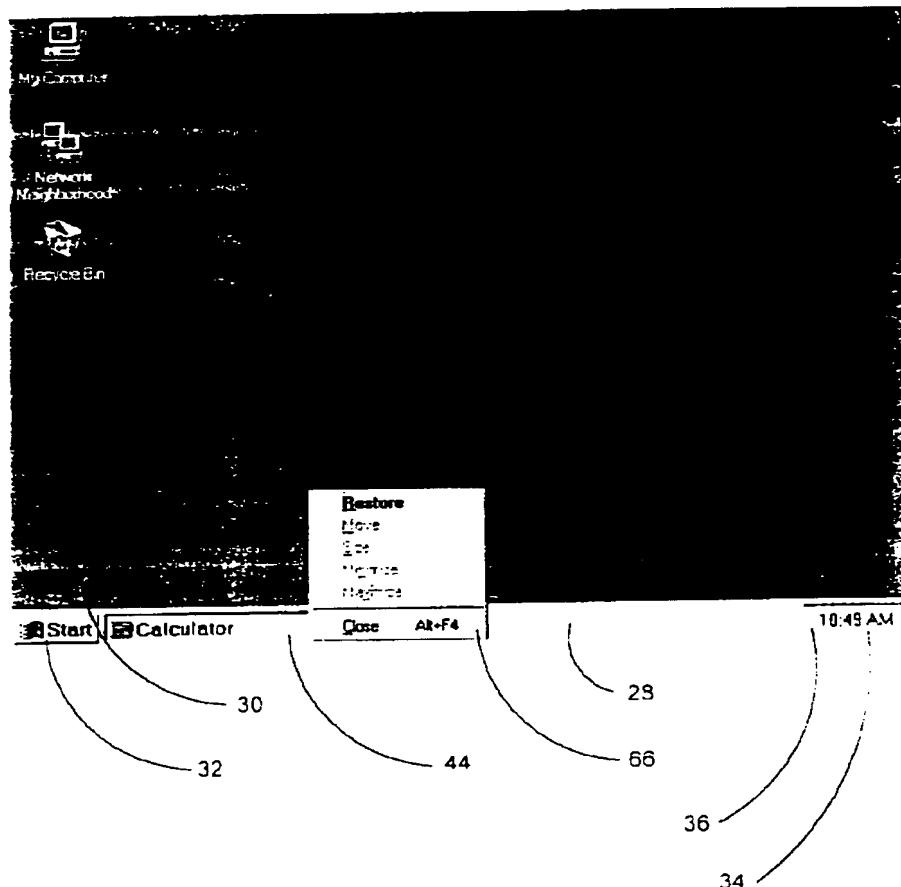
[図11]



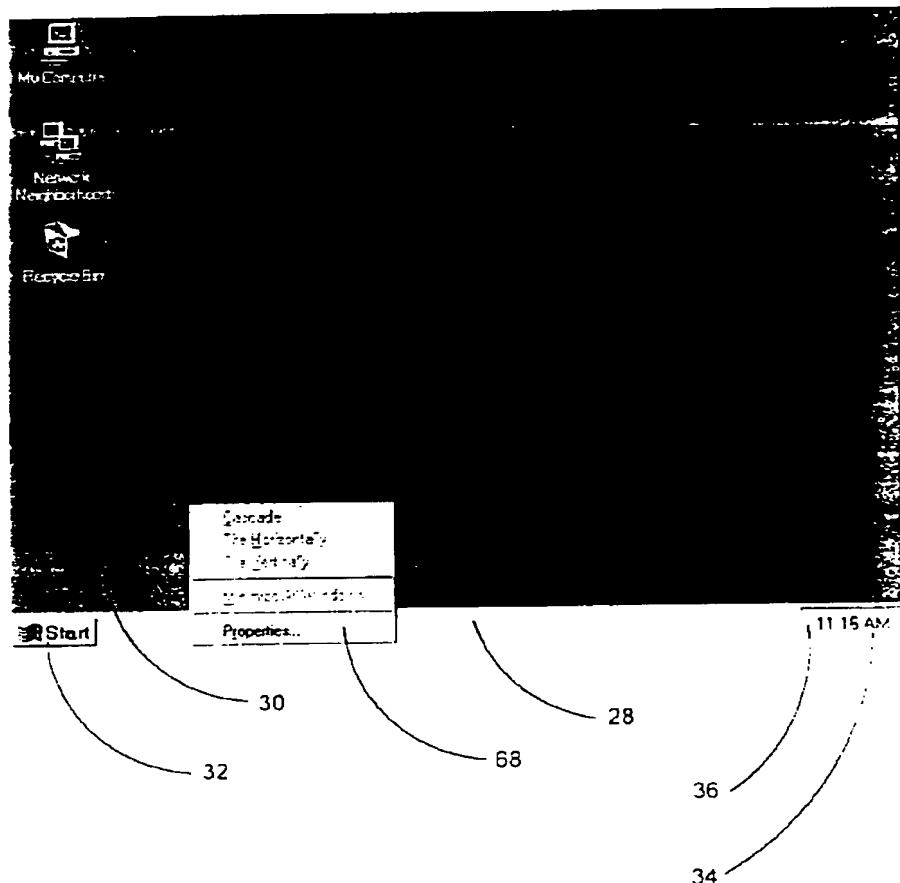
【図12】



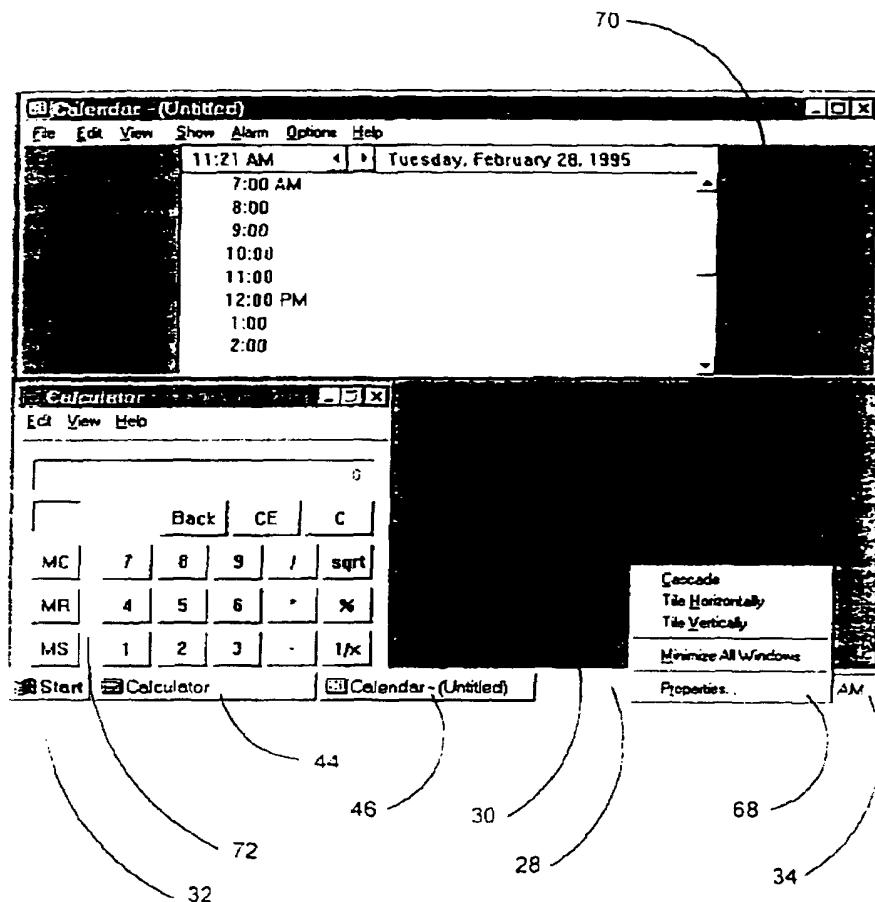
【図13】



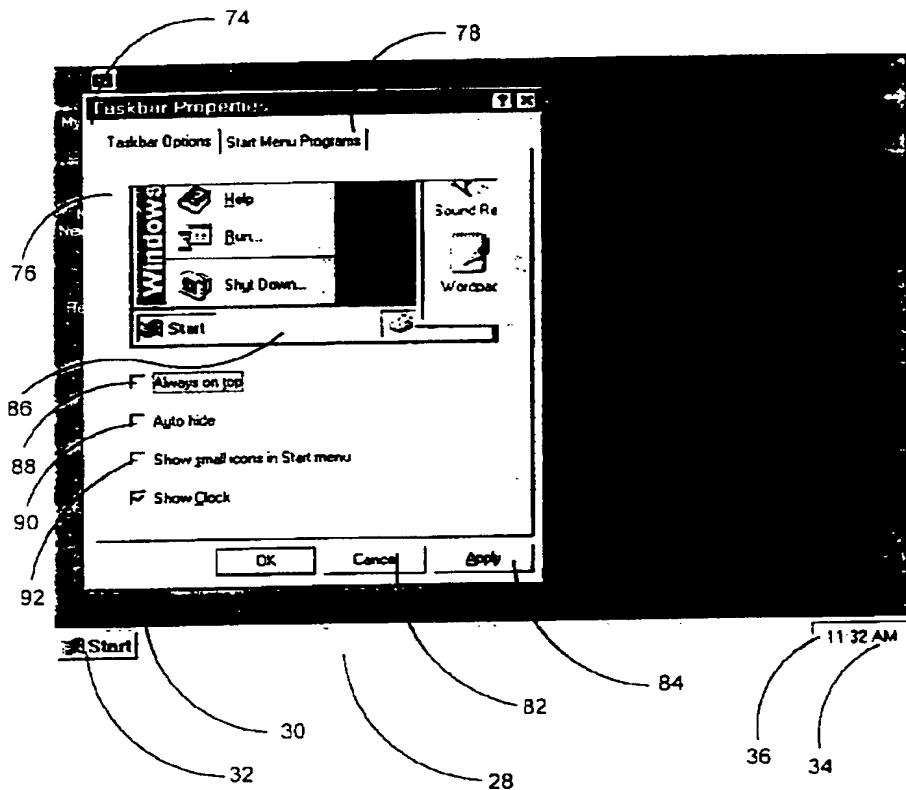
【図14】



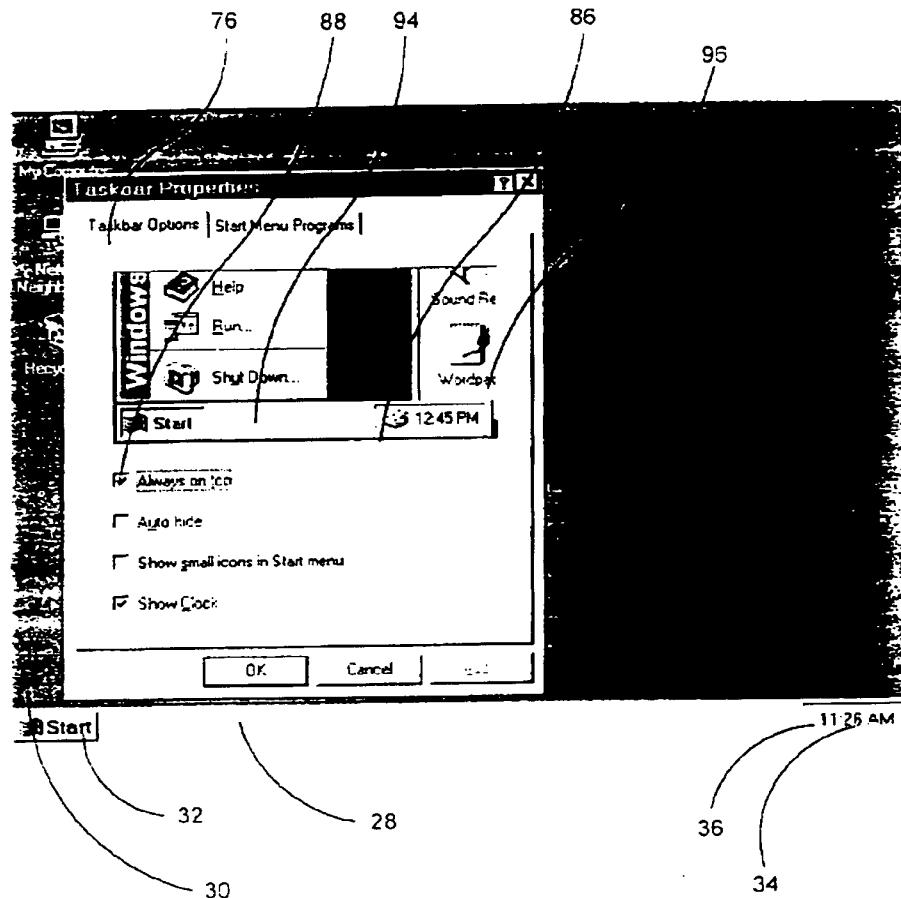
【図15】



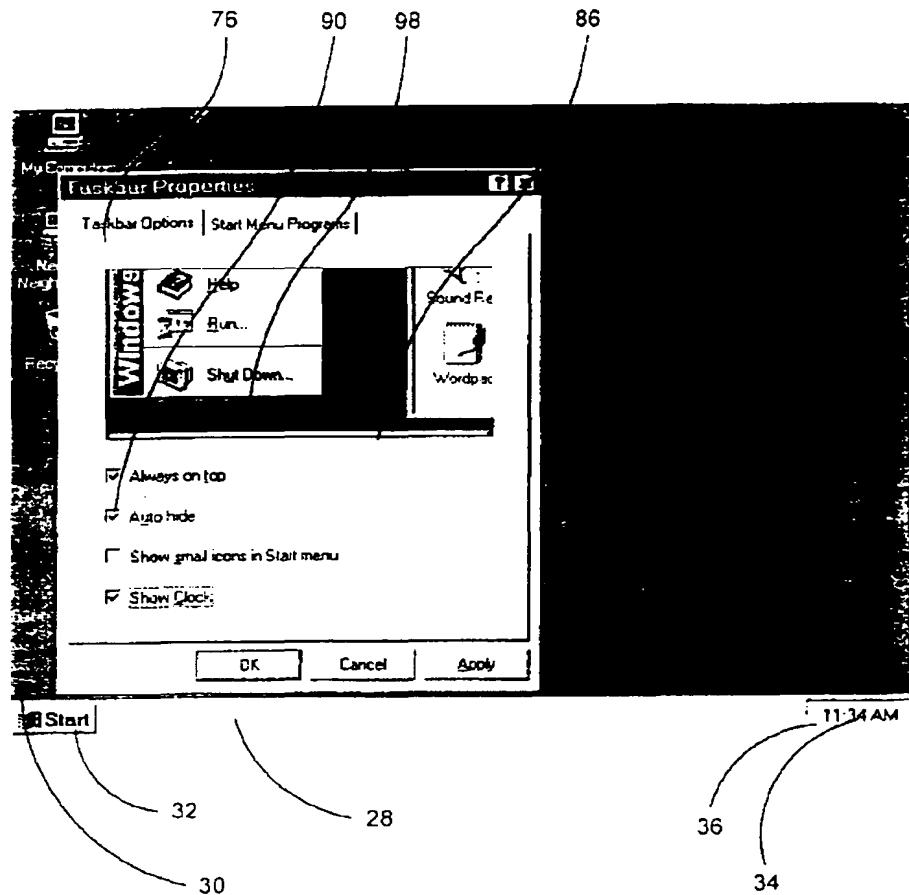
【図16】



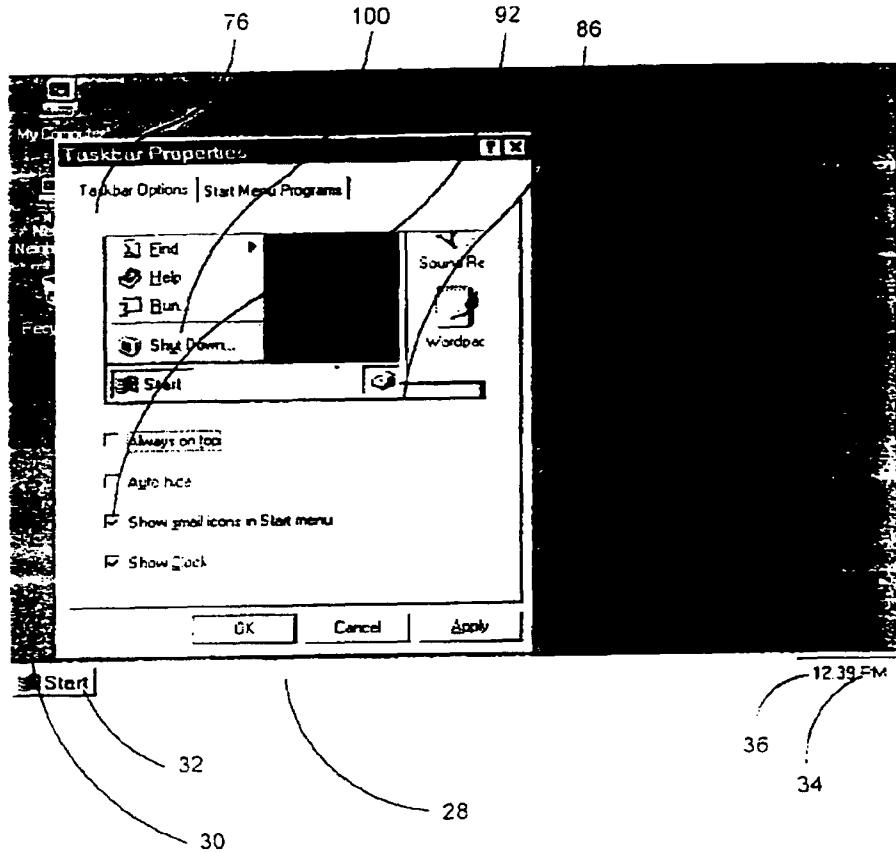
【図17】



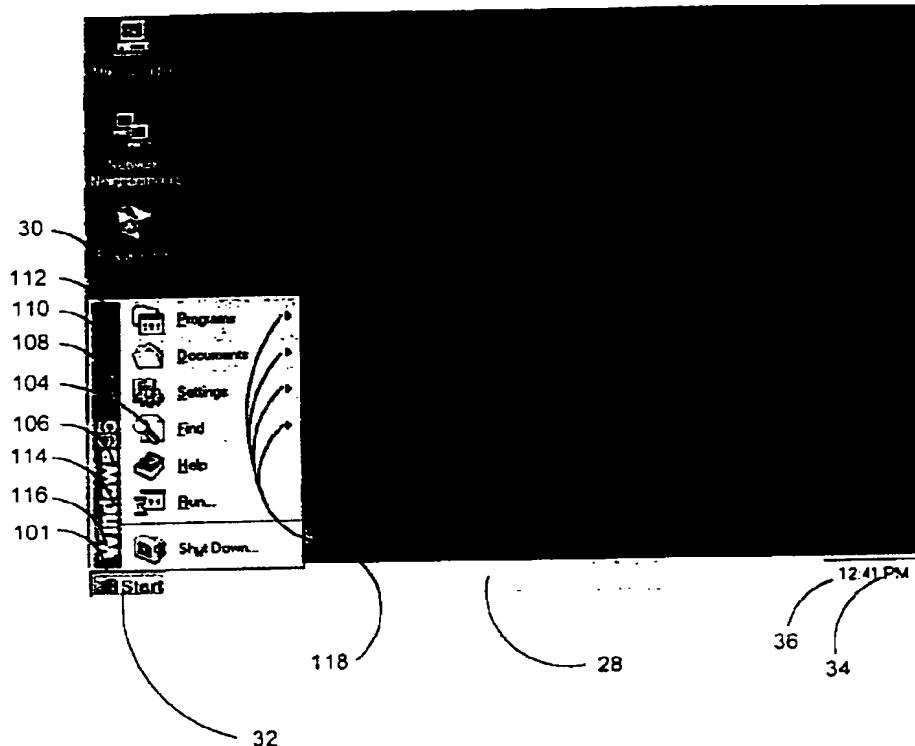
【図18】



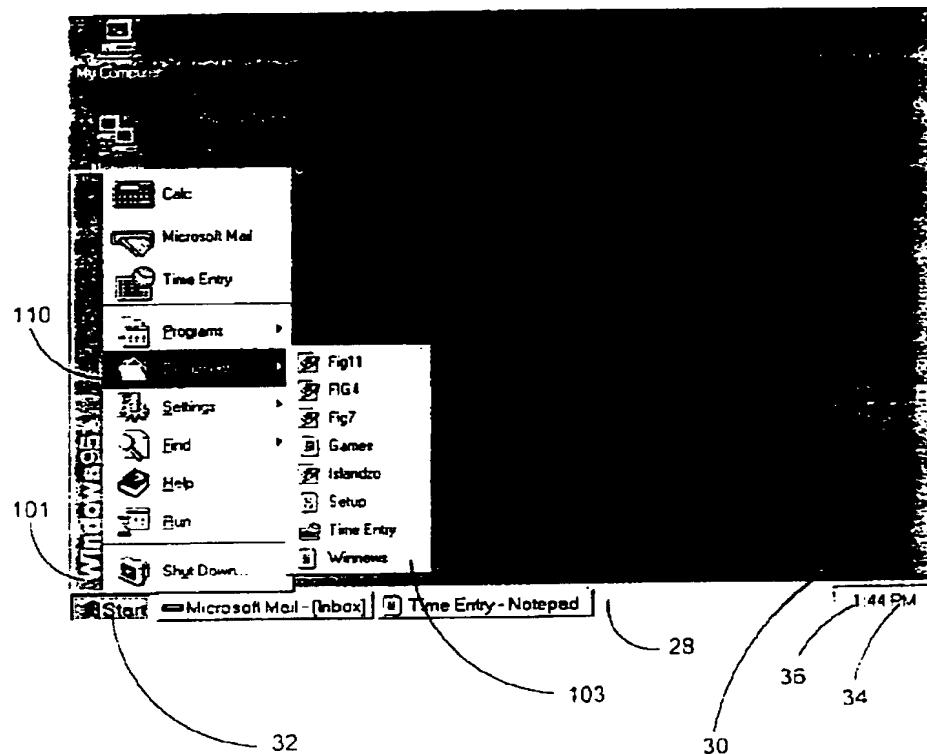
【図19】



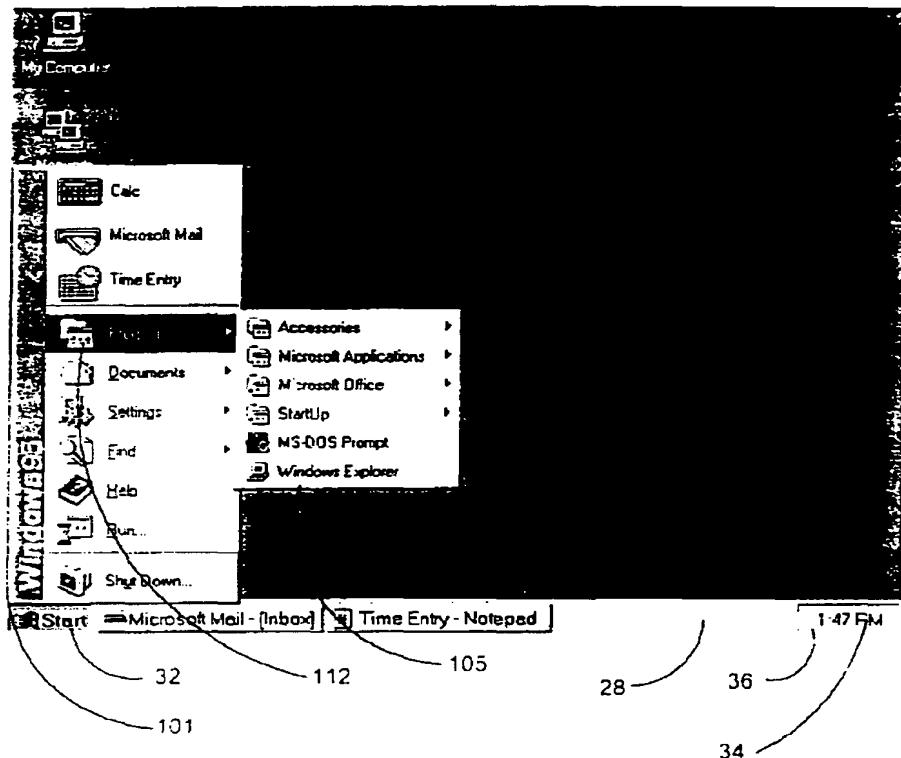
【図20】



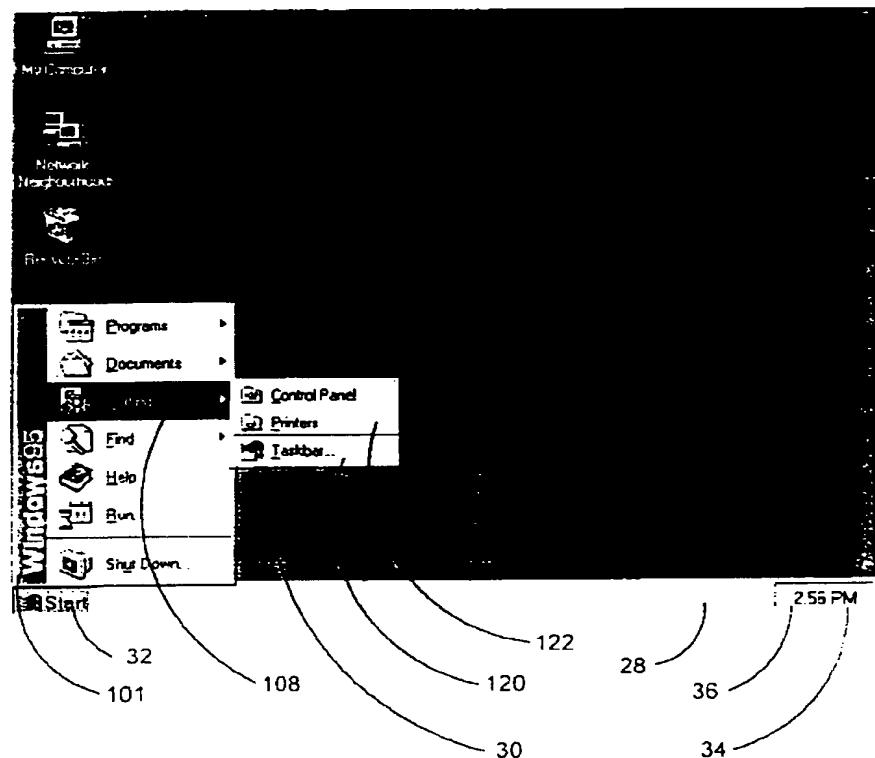
【図21】



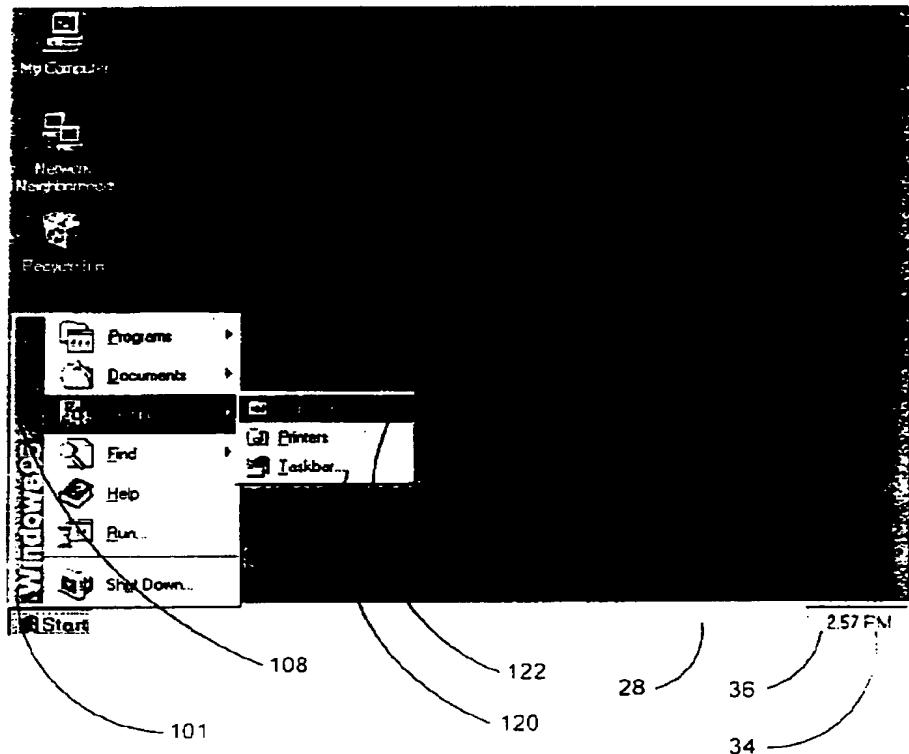
【図22】



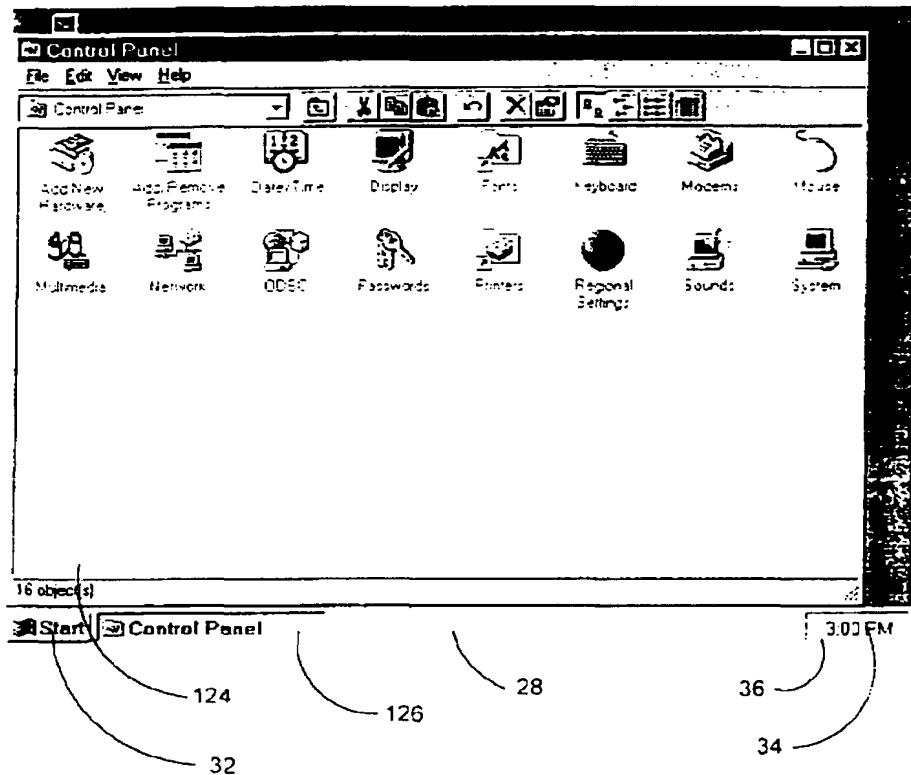
【図23】



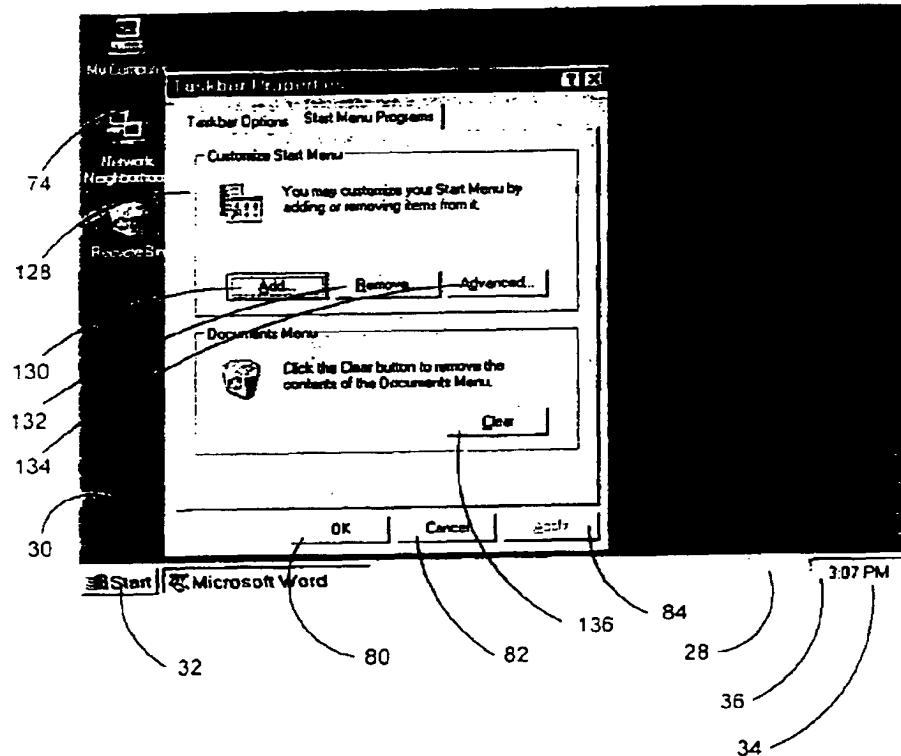
【図24】



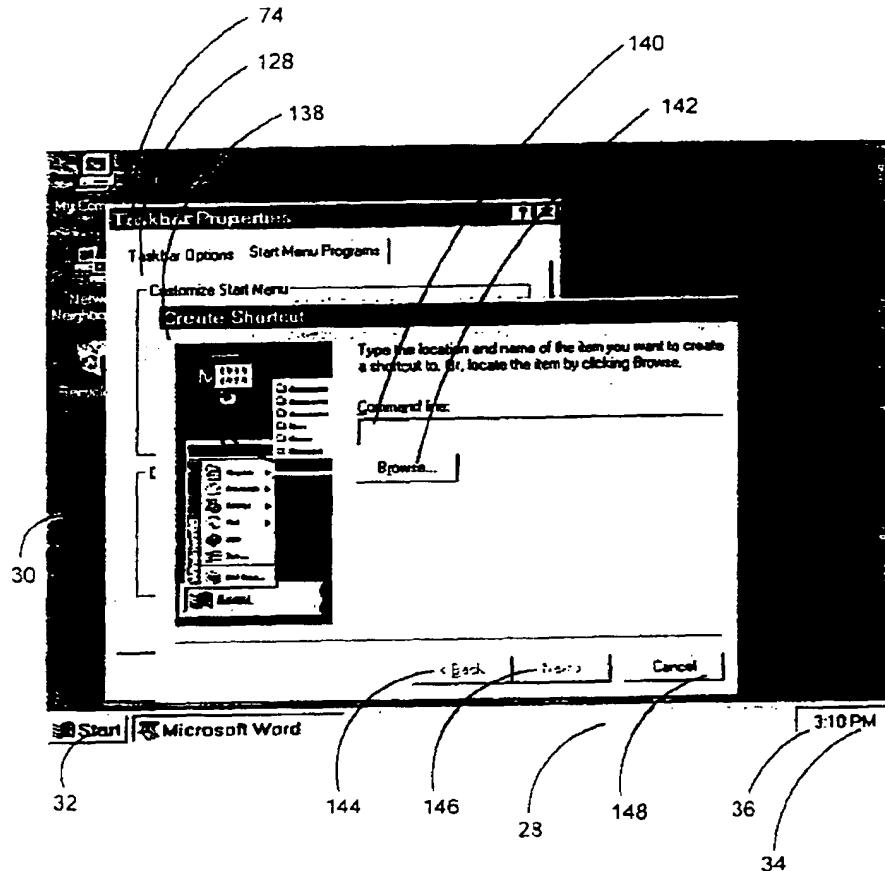
【図25】



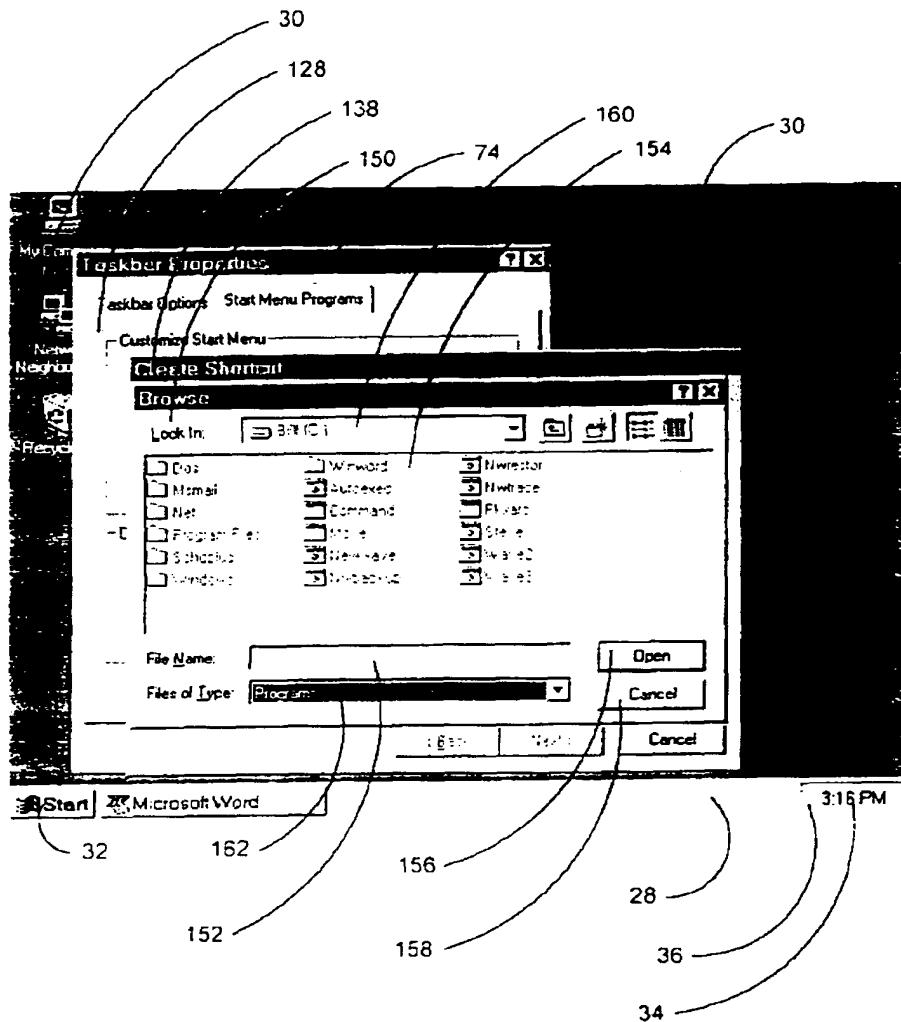
【図26】



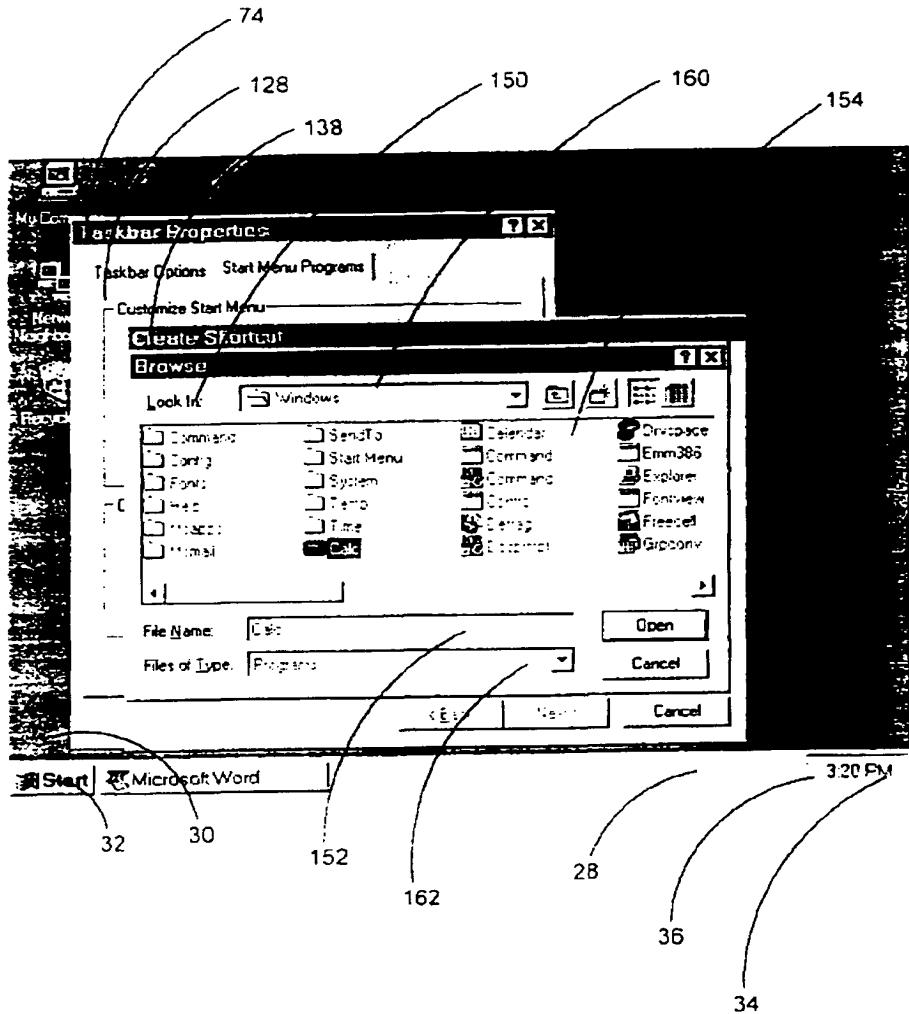
【図27】



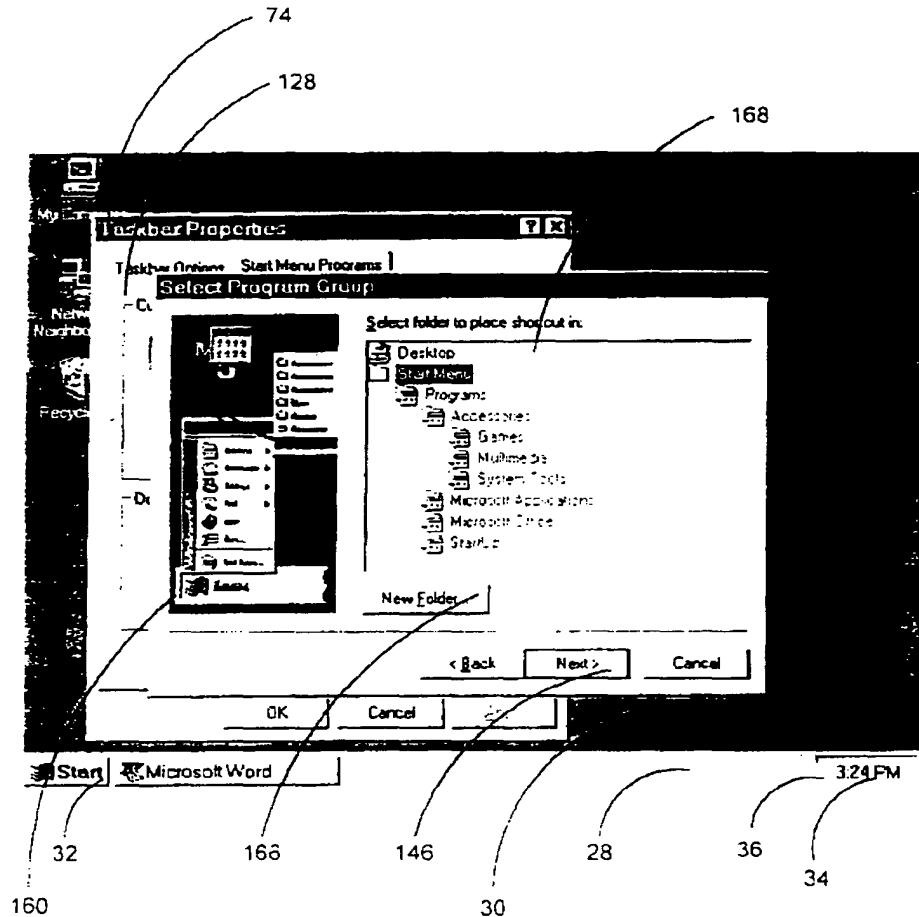
【図28】



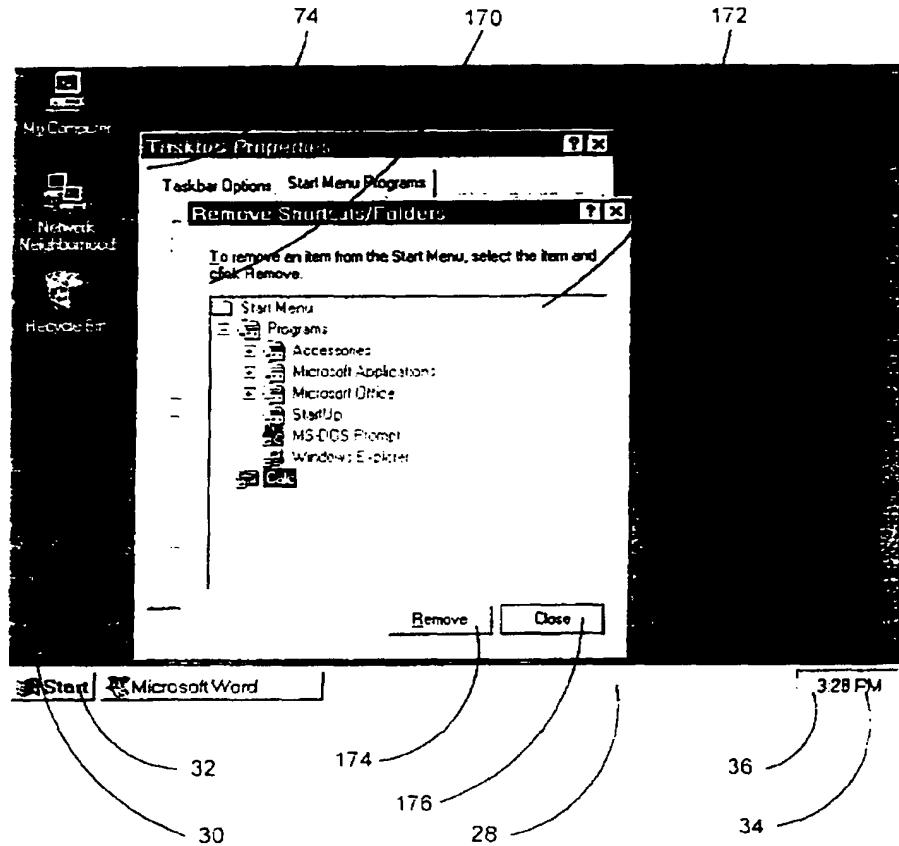
[图29]



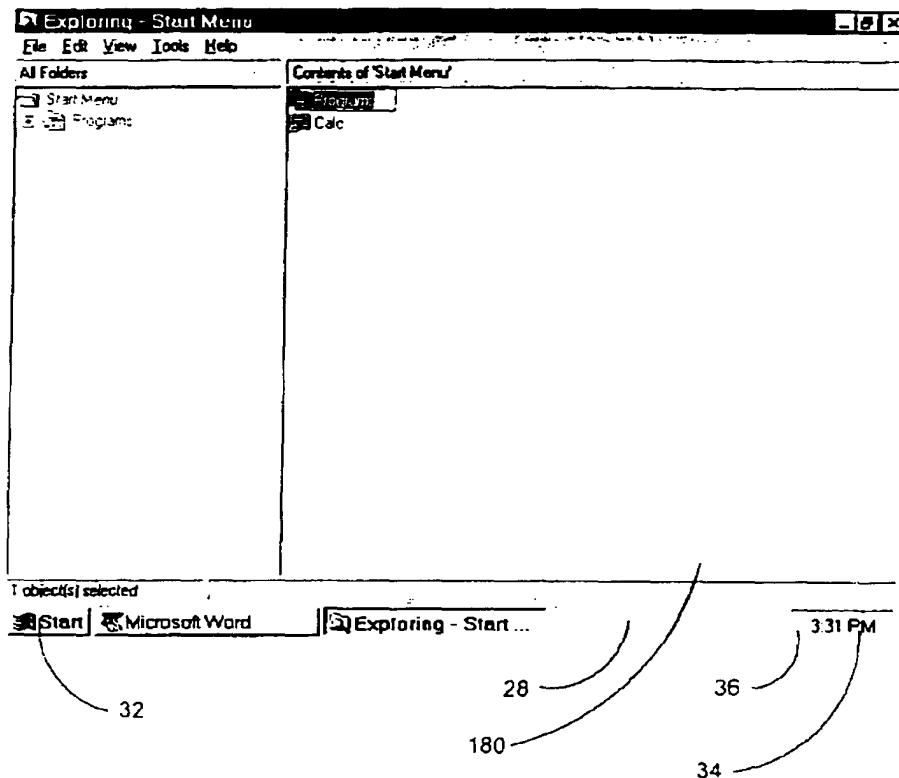
【図30】



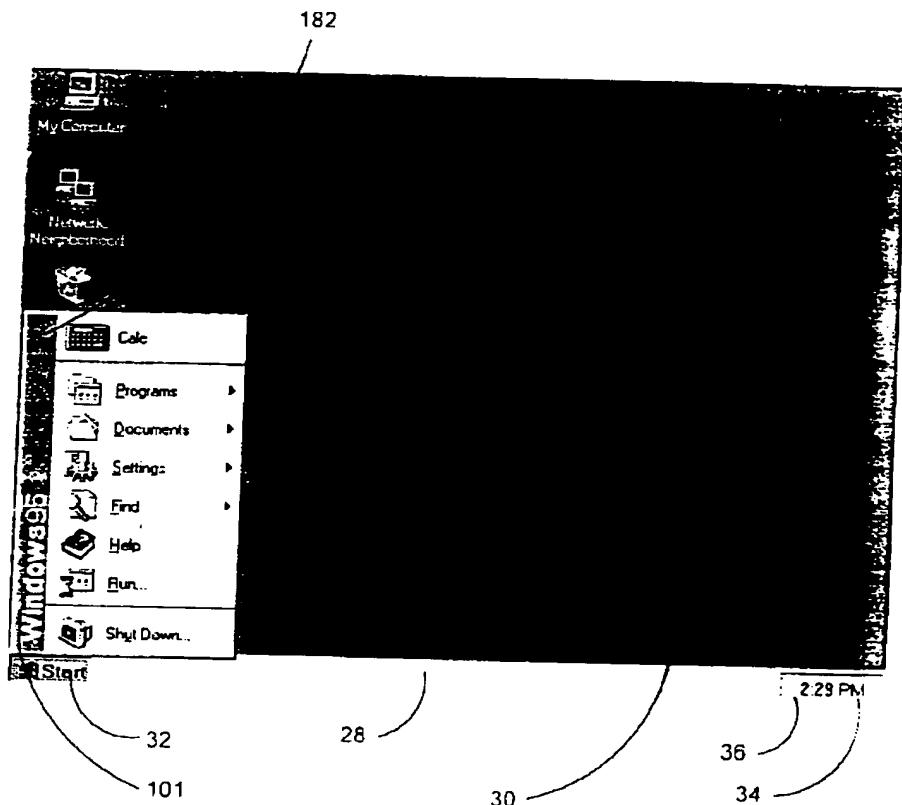
【図31】



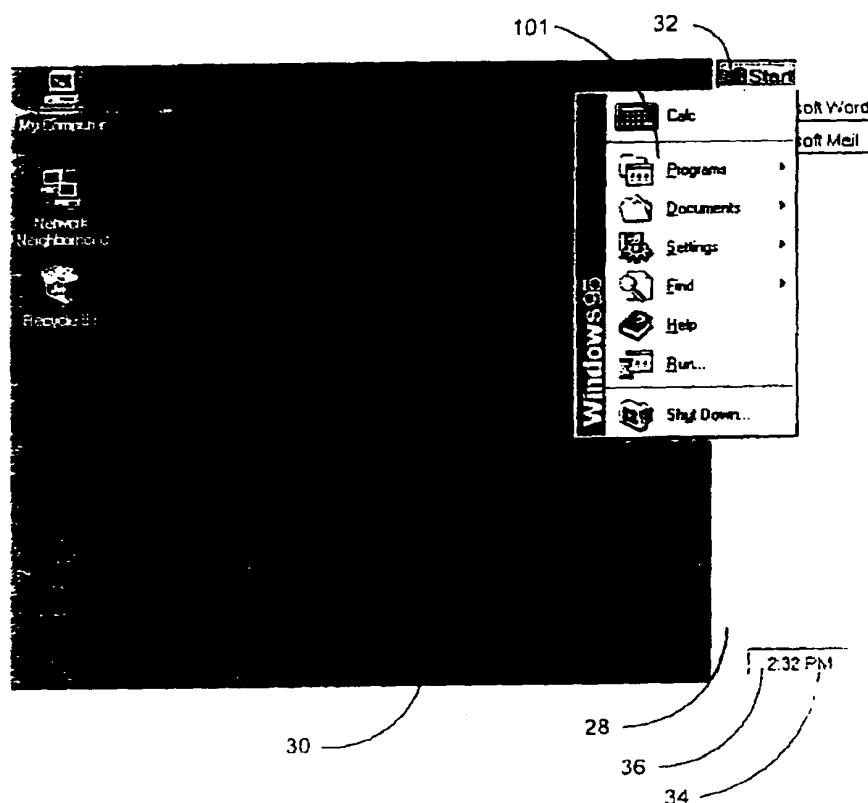
【図32】



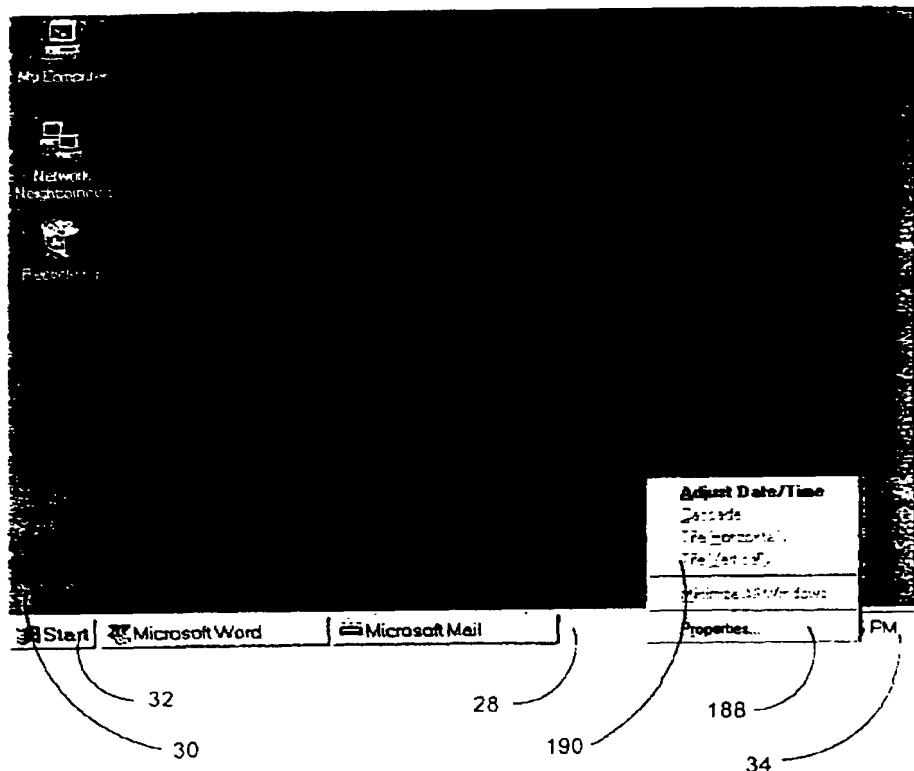
【図33】



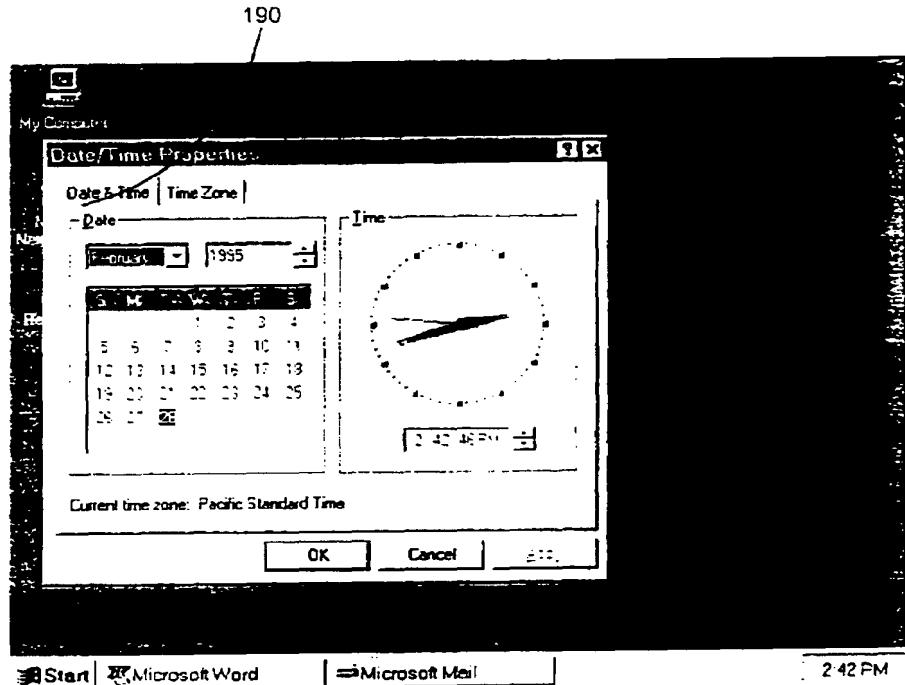
【図34】



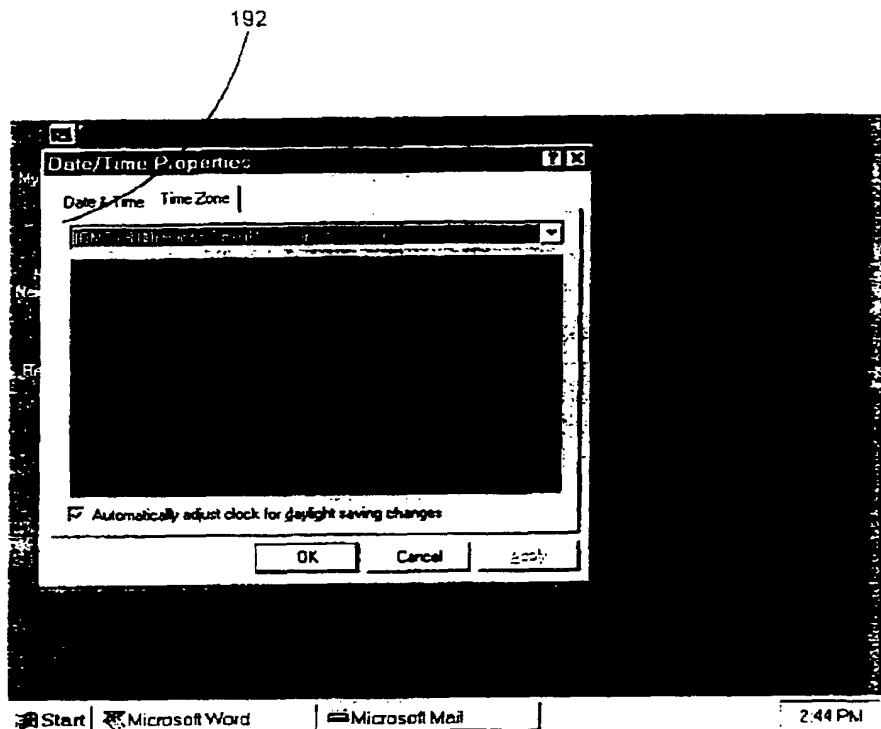
【図35】



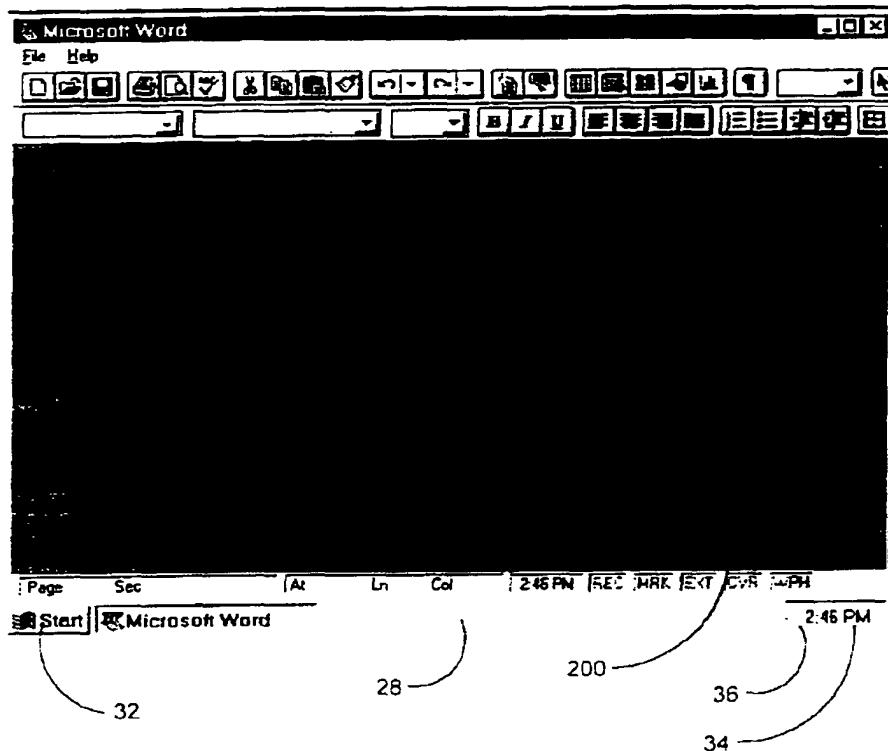
【図36】



【図37】



【図38】



フロントページの続き

(72)発明者 イアン エム エリソン テイラー
アメリカ合衆国 ワシントン州 98105
シアトル ユニヴァーシティー ヴィュー
プレイス ノースイースト 4729
(72)発明者 チー エイチ チュー
アメリカ合衆国 ワシントン州 98052
レッドモンド ワンハンドレッドアンドセ
ブンティエイス アベニュー ノースイ
スト 2001
(72)発明者 ジョセフ ディー ベルフィオーレ
アメリカ合衆国 ワシントン州 98052
レッドモンド イー124 ノースイースト
フォーティース ストリート 17525

(72)発明者 マーク エイ マラムード
アメリカ合衆国 ワシントン州 98119
シアトル ナインス アベニュー ウェス
ト 1508
(72)発明者 ジェフリー エル ボグダン
アメリカ合衆国 ワシントン州 98053
レッドモンド トゥーハンドレッドアンド
シックスス ブレイス ノースイースト
1109
(72)発明者 ケント ディー サリヴァン
アメリカ合衆国 ワシントン州 98008
ベルヴィュー ノースイースト トゥエン
ティシックスス ストリート 16611

【外国語明細書】

1. Title of Invention

TASKBAR WITH START MENU

2. Claims

1. In a computer system having an input device and a video display, a method comprising the steps of:

outputting a user interface element on the video display wherein the user interface element is in a mode such that the user interface element cannot be obscured by windows, said user interface element including a sub-element for each application having an active window in the computer system so as to indicate to the user which applications currently have active windows; and

requesting a selected one of the applications to display a non-minimized window of the selected application by manipulating the sub-element for the selected application in response to the user using the input device.

2. The method of claim 1 wherein the user interface element is a window.

3. The method of claim 1 wherein the computer system includes a mouse having at least one button, said mouse for manipulating a cursor on the video display and wherein the step of requesting the selected one of the applications comprises the step of positioning the mouse cursor to point at the sub-element for the selected application and clicking the button of the mouse.

4. The method of claim 1 wherein the step of requesting the selected one of the applications further comprises the step of changing an appearance of the sub-element for the selected application to indicate that the window for the selected application is visible and non-minimized.

5. The method of claim 1 wherein the step of outputting the user interface element comprises the step of outputting the sub-elements as buttons.

6. The method of claim 1 wherein the step of outputting the user interface element on the video display further comprises the step of outputting a sub-element for accessing a menu, the menu enabling a user to access and launch programs on the system.

7. The method of claim 6 wherein the menu includes an option for accessing a list of most recently opened documents.

8. The method of claim 1 wherein the computer system further comprises an input device and wherein the method further comprises the step of resizing the user interface element in response to a user using the input device.

9. The method of claim 8 wherein the input device is a mouse and wherein the step of resizing the user interface element comprises the step of resizing the user interface element in response to a user dragging and dropping a border of the user interface element with the mouse.

10. The method of claim 8 wherein the step of resizing the user interface element comprises the step of increasing a size of the user interface element.

11. The method of claim 8 wherein the step of resizing the user interface element comprises the step of decreasing a size of the user interface element.

12. The method of claim 1 wherein the step of outputting the user interface element on the video display

further comprises the step of outputting a current time as part of the user interface element on the video display.

13. The method of claim 1 wherein the step of outputting the user interface element on the video display further comprises the step of outputting a taskbar notification area on the video display as part of the user interface element for displaying visual notifications to users.

14. The method of claim 1, further comprising the step of changing the mode of the user interface element to a second mode so that it can be obscured by windows.

15. The method of claim 1, further comprising the step of changing the mode of the user interface element to a third mode so that the user interface element is not visible when a window is placed in a non-minimized state.

16. The method of claim 15, further comprising the step of redisplaying the user interface element in response to a user using the input device.

17. In a computer system having an input device and a video display, a method comprising the steps of:
 providing applications for running on the computer system;
 outputting a taskbar user interface element on the video display, wherein the user interface element is in a mode such that it cannot be obscured by windows;
 displaying visual indicators on the taskbar user interface element for each of the applications that has an active window;
 displaying a non-minimized window of the selected application in response to a user using the input device;

launching a previously closed one of the applications; and

displaying a visual indicator on the taskbar user interface element for the previously closed one of the applications.

18. The method of claim 17, further comprising the step of resizing each of the visual indicators on the taskbar user interface element in response to launching the previously closed one of the applications.

19. The method of claim 17, further comprising the steps of:

opening an additional previously closed one of the applications; and

displaying a visual indicator on the taskbar user interface element for the additional previously closed one of the applications.

20. The method of claim 17 wherein the visual indicators are buttons.

21. A computer system comprising:

(a) a video display;

(b) a processor for running

(i) applications;

(ii) a system resource for displaying a taskbar user interface element for displaying a sub-element for each associated application having an active window, said taskbar user interface element being in a mode wherein windows cannot obscure the taskbar user interface element; and

(c) an input device for manipulating one of the sub-elements to display a non-minimized window for the associated application of the sub-element.

22. The computer system of claim 21 wherein the taskbar user interface element is a window.

23. The computer system of claim 21 wherein the sub-elements are buttons.

24. The computer system of claim 21 wherein the system resource further comprises means for changing appearance of the sub-elements when the windows of their associated applications are non-minimized.

25. The computer system of claim 21 wherein the system resource further comprises means for adding a sub-element to the taskbar user interface element for a new associated application when the new application is launched.

26. The computer system of claim 21 wherein the system resource further comprises means for removing a sub-element from the taskbar user interface element for a selected one of the applications when the window for the selected application is closed.

27. The computer system of claim 21 wherein the system resource further comprises means for resizing the taskbar user interface element.

28. The computer system of claim 27, further comprising an input device for receiving input from a user, and said means for resizing the taskbar user interface element resizing in response to user input from the input device.

29. The computer system of claim 21 wherein the system resource further comprises means for displaying a clock that shows a current time on the taskbar user interface element.

30. The computer system of claim 21 wherein the system resource further comprises means for displaying a start menu element on the taskbar user interface element to gain access to a start menu that provides a user with access to programs.

31. The computer system of claim 21 wherein the input device comprises a mouse having a mouse button for manipulating a cursor on the video display and wherein the system resource comprises means for displaying a non-minimized window for the associated application of a selected one of sub-elements in response to the cursor pointing at the selected sub-element and a user clicking the mouse button.

32. The computer system of claim 21 wherein the taskbar user interface element is anchored to a location on the video display and wherein the computer system further comprises means for changing a location at which the taskbar user interface element is anchored on the video display.

33. In a computer system having a video display and an input device for receiving user input, a method comprising the steps of:

providing a taskbar user interface element that is anchored to a location on the video display, said taskbar user interface element including visual elements that identify each application that has an active window and that may be used to make visible non-minimized windows for the applications; and

providing a means for moving the taskbar user interface element in response to the user using the input device so that the taskbar user interface element is anchored to another location on the video display; and

using the means for moving the taskbar user interface element to anchor the taskbar user interface element to a new location on the video display.

34. In a computer system having an input device for receiving user input, a video display for displaying video output and a processor running an operating system, a method comprising the steps of:

displaying with the operating system a taskbar user interface element on the video display that includes a visual element for accessing a start menu and an indicator of currently active tasks;

displaying the start menu in response to a user using the input device to manipulate the visual element for accessing the start menu, said start menu including a documents option; and

in response to a user selecting the documents option with the input device, displaying a list of most recently opened documents, which the user may access via the list.

35. In a computer system having an input device for receiving user input, a video display for displaying video output and a processor running an operating system, a method comprising the steps of:

displaying with the operating system a taskbar user interface element that includes a visual element for accessing a start menu and an indicator of currently active tasks;

providing means for configuring the start menu to enable a user to add items to the start menu;

using the means for configuring the start menu to add a new item to the start menu at a request of the user; and displaying the start menu with the new item, said start menu also including items for a user to access programs and documents.

36. In a computer system having an input device for receiving user input and a video display for displaying video output, a method comprising the steps of:

displaying a taskbar user interface element on the video display, said taskbar user interface element including visual indicators of currently active tasks;

providing a means for displaying a context menu for the taskbar user interface element, said context menu including items for determining how windows are displayed relative to each other and a selected item for undoing a last choice of one of the items for determining how windows are displayed relative to each other;

using the means for displaying the context menu to display the context menu; and

in response to the user choosing the selected item, undoing an effect of a last choice of one of the items for determining how windows are displayed relative to each other.

37. In a computer system having a mouse for receiving user input and a video display for displaying video output, a method comprising the steps of:

displaying an object on the video display;

displaying a taskbar user interface element on the video display, said taskbar user interface element including a visual indicator for each application having an active window;

dragging the object to be positioned over a selected one of the visual indicators on the taskbar user interface element in response to the user using the mouse and remaining positioned over the selected visual indicator for a non-zero threshold period of time; and

in response to the object being positioned over the selected visual indicator for the threshold period of time, opening the window for the application of the selected visual indicator.

38. The method of claim 37, further comprising the step of dropping the object in window for the application of the selected visual indicator in response to the user using the mouse.

39. In a computer system having an input device and a video display, a method comprising the steps of:

displaying a user interface element on the video display when no windows for application programs are currently in a non-minimized state, said user interface element including a visual indicator for each application that has an active window to identify the associated application and for placing the windows of the associated applications in a visible non-minimized state; and

hiding the user interface element from being displayed on the video display when a window of one of the applications is in a visible non-minimized state.

40. The method of claim 39, further comprising the step of redisplaying the user interface element when a window for one of the applications is placed in a non-minimized state in response to a user using the input device.

41. The method of claim 39 wherein the user positions a cursor in a predefined area of the video display to cause the user interface element to be redisplayed.

42. The method of claim 41, further comprising the step of again hiding the user interface element when the cursor is no longer in the predefined area of the video display.

3. Detailed Description of Invention

Technical Field

This invention relates generally to data processing systems and, more particularly, to the use of a taskbar having a start window in a data processing system.

Background of the Invention

The "MICROSOFT" WINDOWS, Version 3.1, operating system, sold by Microsoft Corporation of Redmond, Washington, provides users with a window-based graphical user interface. In the environment provided by this operating system, application programs have associated windows that users make visible in a non-minimized form to use the applications. Typically, an application has an associated icon. The user makes the window for the application visible in a non-minimized form by positioning a mouse cursor to point at the icon for the application and then double clicking a designated mouse button. Unfortunately, double clicking the mouse button is difficult for some users. For example, some users have difficulty in clicking the mouse button twice with sufficient speed for the system to interpret their actions as double clicking. Furthermore, certain users inadvertently move the mouse during the double clicking so that their actions are not properly interpreted.

Another difficulty encountered in the graphical user interface of the "MICROSOFT" WINDOWS, Version 3.1 operating system is that visible non-minimized windows and minimized windows (*i.e.*, icons) may be obscured by other non-minimized windows or icons. As a result, it is often difficult for novice users to know what applications are currently active. Further, many novice users are confused by one window obscuring another. These users do not

understand what has happened to the window that has been obscured.

Many novice users are also confused by the hierarchy provided by this environment. The top of the hierarchy is the program manager which displays icons for each of the program groups that are available. The next level of the hierarchy is formed by the program groups. To gain access to the programs in a program group, a user typically opens a window associated with the program group. The programs group window displays icons for each of the applications that is available in the program group. The user may then open any of these programs via a file menu or by double clicking on the program icon. Traversing this hierarchy is quite difficult for many users and many users get lost in the hierarchy so that they do not know how to properly navigate.

Other complications with this conventional approach are that users do not know how or where to obtain help information. Further, users have difficulty opening documents and switching programs.

Summary of the Invention

In accordance with a first aspect of the present invention, a method to practice a computer system having an input device and a video display. In this method, a user interface element is output on the video display. The user interface element is in a mode such that it cannot be obscured by windows. The user interface element includes a sub-element for each application that has an active window in the computer system. The sub-elements indicate to the user which applications currently have active windows. A request is made to display a non-minimized window of a selected one of the applications by manipulating the sub-element in response to the user using the input device.

In accordance with another aspect of the present invention, a method is practiced in an environment wherein applications are being run on a computer system. A task for a user interface element is output on a video display in a mode such that it cannot be obscured by windows. Visual indicators are displayed on the task bar user interface element for each of the applications that has an active window. A non-minimized window for a selected one of the applications is displayed in response to the user using an input device. A previously closed application is then launched and a visual indicator is displayed on the taskbar user interface element for the previously closed application.

In accordance with a further aspect of the present invention, the computer system includes a video display and a processor. The processor is used for running applications and a system resource for displaying a taskbar user interface element. The taskbar user interface element displays a sub-element for each associated application that has an active window. The taskbar user interface element is in a mode where windows cannot obscure the taskbar user interface element. The computer system also includes an input device for manipulating one of the sub-elements to display a non-minimized window for the associated application of one of the sub-elements.

In accordance with an additional aspect of the present invention, a task for a user interface element is provided on a video display such that it is anchored to a location on the video display. The taskbar user interface element includes visual elements that identify each application that has an active window. These visual elements may be used to make visible non-minimized windows for the applications. A means for moving the taskbar user interface element in response to a user using an input device is provided so that a taskbar user interface

element may be anchored to another location on the video display. This means is moved to anchor the taskbar user interface element to a new location on the video display.

In accordance with yet another aspect of the present invention, a taskbar user interface element is displayed on the video display by an operating system. The taskbar user interface element includes a visual element for accessing a start menu and an indicator of currently active tasks. The start menu is displayed in response to a user using an input device to manipulate the visual element for accessing the start menu. The start menu includes a documents option. In response to a user selecting the documents option with the input device, a list of the most recently opened documents which the user may access is displayed.

In accordance with a further aspect of the present invention, a task user interface is displayed to include a visual element for accessing a start menu and an indicator of currently active tasks. A means for configuring the start menu is used to add a new item to the start menu at a request of a user. The start menu is then displayed with the new item. The start menu also includes items for a user to access programs and documents.

In accordance with an aspect of the present invention, a taskbar user interface element is displayed on a video display. The taskbar user interface element includes visual indicators of currently active tasks. A means for displaying a context menu for the taskbar user interface element is used. The context menu includes items for determining how windows are displayed relative to each other. The context menu also includes a selected item for undoing a last choice of one of the items for determining how windows are displayed relative to each other. In response to a user choosing the selected item, the affect of a last choice of one of the items for

determining how windows are displayed relative to each other is undone.

In accordance with an aspect of the present invention an object is displayed on a video display. A taskbar user interface element is also displayed on the video display. The taskbar user interface element includes a visual indicator for each application that has an active window. The object that is displayed on the video display is dragged to be positioned over a selected one of the visual indicators on the taskbar user interface element in response to the user using a mouse. The object remains positioned over the selected visual indicator for a threshold period of time. In response to the object being positioned over the selected visual indicator for the threshold period of time a window for the application of the selected visual indicator is opened.

In accordance with a further aspect of the present invention, a user interface element is displayed on a video display when no windows for application programs are currently visible in a non-minimized state. The user interface element includes a visual indicator for each application that has an active window to identify the associated application and for opening windows of the associated applications. The user interface element is hidden from being displayed on the video display when a window of one of the applications is opened.

Detailed Description of the Invention

The preferred embodiment of the present invention provides a taskbar and start menu for overcoming many of the difficulties encountered by conventional systems. The taskbar is a consistently visible user interface element that informs a user which tasks are currently active. The taskbar is constructed so that it is not obscurable by open windows, unless the user so selects to obscure the taskbar. The taskbar also serves as a point for displaying currently active task windows. The taskbar includes a button for each active task. Users merely need to single click on a taskbar button provided for a task to display the task window. These button may also be used to switch tasks. The taskbar remains

anchored at a fixed location on the user interface. This fixed anchor location may, however, be changed by a user. The taskbar also includes a start menu button which serves as a centralized location for accessing programs via the user interface. The start menu includes menu items that allow a user to readily access desired programs, documents or system controls. The start menu is configurable, as will be described in more detail below.

Figure 1 is a block diagram of a computer system 10 that is suitable for practicing the preferred embodiment of the present invention. The computer system includes a central processing unit (CPU) 12, which has access to several peripheral devices, including a mouse 14, a keyboard 16 and a video display 18. The mouse 14 includes at least a left mouse button and right mouse button. The CPU 12 also has access to memory 20 and secondary memory 22. The memory 20 holds copies of applications 24 and an operating system 26. In the preferred embodiment of the present invention, the taskbar and start menu are provided by the shell of the operating system 26. Nevertheless, those skilled in the art will appreciate that the present invention is not limited to being practiced in an operating system. Rather, the present invention may also be implemented through utilities or other system resource. For purposes of the discussion below, it is assumed that the operating system 26 is the "MICROSOFT" WINDOWS 95 operating system from Microsoft Corporation. Those skilled in the art will appreciate that the present invention may also be practiced with other operating systems.

The taskbar is implemented by the operating system 26 as a window. With the operating system 26, a window is always created based upon a window class. The window class identifies a window procedure that processes messages that are sent to the window. The operating system 26 supports a RegisterClass() function that enables an application to register window classes. The window procedure is a function that receives and processes messages that are sent to the window. The operating

system 26 is a message-based operating system such that operations are achieved by passing messages between windows.

Figure 2 shows an example of the graphical user interface provided by the operating system 26 of the preferred embodiment of the present invention. A taskbar 28 is positioned at the bottom of a virtual desk top 30 on the video display 18 (Figure 1). In the example shown in Figure 2, the taskbar 28 extends from the left-hand side of the user interface to the right-hand side of the user interface and is anchored to the bottom edge of the video display. The taskbar 28 has a uniform width throughout its length. The taskbar 28 includes a start menu button 32 for gaining access to the start menu. The start menu will be discussed in more detail below. The taskbar 28 also includes a clock 34 and taskbar notification area 36. The clock 34 displays a current time. The taskbar notification area 36 may display icons that serve as visual notifications of events or to convey status information. The taskbar notification area is described in more detail in co-pending application entitled "System Provided Visual Notification Area," Serial No. _____, which is assigned to a common assignee with the present invention and was filed on even date herewith. The start menu button 32 and the clock 34 maintain fixed positions within the taskbar 28 as long as the taskbar remains anchored at a given location.

The taskbar buttons and start menu button 32 are implemented as tab controls, such as described in co-pending application entitled "System Provided Child Window Controls," Serial No. _____, which is assigned to a common assignee with the present application and was filed on even date herewith. Controls are child windows that reside within parent windows and send notification message to parent windows when events, such as input from a user, occur within the child window. In particular, a child

window processes a mouse and keyboard messages and notifies the parent window when a state of child window is changed. This enables the child window to act as a kind of input device of the parent window.

As mentioned above, the taskbar 28 serves as a centralized location for identifying all of the currently active tasks within the system. Figure 3 shows an example of the appearance of the taskbar 28 when a word processing program and an electronic mail program are currently active. The taskbar 28 indicates the tasks that are active by displaying taskbar buttons like button 38 and button 40 shown in Figure 3. The borders of the buttons 38 and 40 are drawn as raised to provide a three-dimensional effect and to indicate that the window for the task is not opened. A user displays an active task window from the taskbar 28 by positioning a mouse cursor 41 to point at a taskbar button using the mouse 14 or other suitable input device and then single clicking the left mouse button to open the window associated with the task to display it in a visible non-minimized state. For example, suppose that only the word processing program was currently active and the user clicked on the left mouse button of the mouse 14 while the mouse cursor 41 was positioned at the button 38 for the word processing program, window 42 for the word processing program is opened as shown in Figure 4. In addition, the taskbar button 38 for the word processing program is redrawn to indicate that it is depressed. Thus, the user is provided with a visual cue that a window, that is associated with the task, is currently active.

From a programming perspective, the following steps take place to display a window for a task that has a button on a taskbar. Initially, the taskbar button, as a child window control, receives a mouse message when the mouse cursor is positioned to point at the taskbar button. The taskbar button then receives a mouse message

indicating that the left mouse button has been clicked. This mouse message is passed on to the parent taskbar process, which is responsible for managing and displaying the taskbar 28. The taskbar 28 receives the mouse messages and, in response, decides to open the word processing window as shown in Figure 4. In addition, the child window control changes its appearance to indicate that the button is depressed.

The size of the taskbar buttons varies depending upon the amount of space that is available on the taskbar 28. Figure 3, shows taskbar buttons 38 and 40 with their maximum permissible size. If, however, additional tasks become active, taskbar buttons for these additional tasks must be added to the taskbar 28. In many instances, in order to adequately display all of the active tasks, the taskbar buttons must be redrawn to a new size. For example, Figure 5 shows an example wherein a schedule program and a calculator program have also become active. Thus, the taskbar buttons for these newly activated tasks must be displayed on the taskbar 28. In order to adequately display all of the taskbar buttons for all of the active tasks, the size of all of the taskbar buttons is changed. Hence, taskbar buttons 38, 40, 44, and 46 as shown in Figure 5 are of smaller width than the taskbar buttons 38 and 40 shown in Figure 3.

In order to determine a new size for the taskbar buttons when resizing is necessary, the taskbar calculates the available space within the taskbar 28 for displaying taskbar buttons. The taskbar then determines the number of buttons that must be displayed and divides the available space by the number of taskbar buttons to be displayed. The taskbar then determines the appropriate size for the taskbar buttons based on these calculations. When the taskbar buttons are displayed in an instance where the taskbar 28 is anchored to the bottom edge of the video display 18, the buttons are displayed in a left-

justified fashion. The operating system 26 provides mechanisms for redrawing each of the taskbar buttons as child window controls. Essentially, redrawing the taskbar buttons involve resizing each of the windows associated with the taskbar buttons.

A user may move the interior border of the taskbar 28 to increase or decrease the size of the taskbar. The interior border is the border that faces in towards the rest of the graphical user interface. In the example shown in Figure 7, the interior border 54 is the top border of the taskbar 28. A user may reposition the interior taskbar border at discrete increments (i.e., discrete increments of the minimal height shown in Figure 7) by performing the steps shown in Figure 6. Initially, a user positions the cursor to point at the interior taskbar border (step 48). The conventional pointing arrow cursor (like cursor 41 shown in Figure 3) then changes to a double-arrowed cursor to indicate the position to move the interior border of the taskbar. The user then depresses the left mouse button, drags the taskbar to the desired size (step 50) and drops the interior border by releasing the left mouse button to resize the taskbar (step 52). As shown in Figure 7, a phantom border 56 is displayed during the drag and drop operation to indicate the proposed new interior border position given the current position of the double arrowed cursor 58.

As mentioned above, the interior border may only be moved to discrete positions that are determined based upon the current position of the double arrowed cursor 58. Moreover, the interior border may only be moved to a maximum position wherein if dropped, the taskbar would occupy at most half of the graphical user interface. Figure 8 shows an example of the taskbar 28 as it appears after it has been expanded by dragging the interior border to double the size of the taskbar as depicted in the earlier figures. As the taskbar 28 is a window, the

resizing of the taskbar is handled like the resizing of other windows. The taskbar process in the operating system 26 which is responsible for managing and displaying the taskbar 28 receives the mouse messages regarding the drag and drop operation and resizes the taskbar window accordingly.

As was discussed above, the taskbar may be moved to be anchored at different edges of the video display 18. To move the taskbar 28 to a new anchoring edge, a user performs the steps shown in Figure 9. In particular, a cursor is positioned to point at a location in the taskbar 28. The left mouse button is then depressed and the taskbar is dragged to the desired edge at which it is to be anchored (step 62 in Figure 9). The taskbar is then dropped so that it is redrawn and anchored at the appropriate edge (step 64). The system logically divides the screen into quadrants. Each quadrant is associated with an edge. The taskbar is anchored to the edge associated with the quadrant in which the mouse cursor hot spot lies at the time of the drop. Figure 10A shows the taskbar of Figure 8 as it appears when dragged to the right side edge of the video display 18. It is worth noting that buttons 38, 40, 44, 46 and 66 are redrawn and repositioned based upon where the taskbar is anchored. The taskbar buttons are drawn so as to extend horizontally for each anchoring edge. Start menu button 34 and clock and taskbar notification area are also repositioned accordingly. Figure 10B shows an example of the appearance of the taskbar 28 when it is dragged to the top of the video display 18, and Figure 10C shows an example of the appearance of the taskbar 28 when it is dragged to be anchored to the left edge of the video display 18.

When a user positions a mouse cursor to point at one of the taskbar buttons and then depresses the right mouse button, a context menu is displayed. Figure 11 shows an example of a context menu 66 that is displayed

when the mouse cursor is positioned to point at taskbar button 44 and the right mouse button is depressed. The context menu 66 allows a user to select from different context-sensitive operations that may be performed on the window associated with the taskbar button. For example, the context menu 66 includes an option to "Close" the window that is associated with taskbar button 44.

A context menu 68 (Figure 12) is also provided for the taskbar 28. This taskbar 28 is displayed when a user positions a mouse cursor to point at the taskbar and depresses the right mouse button. The context menu 68 includes a "Cascade" option for cascading open windows. The context menu 68 for the taskbar also includes a "Tile Horizontally" option for horizontally tiling open windows. A "Tile Vertically" option is also provided in the context menu 68 to vertically tile open windows. These operations are performed using conventional techniques. The "Minimize All Windows" option minimizes all currently open windows so that they appear as taskbar buttons on the taskbar 28. The operating system 26 includes the notion of a definable work area that is usable to the user. The windows are minimized so as to not lie in the work area. Hence, they are not visible to the user. The taskbar is also informed of the minimization request and knows to update the taskbar buttons for each of the windows so that the buttons appear raised.

The context menu 68 may also include an "Undo Tile" option for undoing the previously selected windows tiling option as shown in Figure 13. For the example case shown in Figure 13, a window 70 for a calendar program and a window 72 for a calculator program are currently open. The "Tile Vertically" option has been selected. The windows may be displayed as they were previously displayed before the request to tile them vertically was made by choosing the "Undo Tile" option in the context menu 68. The system achieves this undo tiling effect by recording

the previously selected windows tiling option and reinstituting that tile option after the undo is requested.

A final option on the taskbar context menu 68 is the "Properties" option, which may be selected to display property sheets for the taskbar 28 and start menu. Figure 14 shows what is displayed when the "Properties" option is selected from the taskbar or context menu 68. A frame window 74 labeled as "Properties for Taskbar" is displayed. The frame window 74 is used to hold property sheets, like a frame. A property sheet 76 for "Taskbar Options" is displayed in the frame window 74, along with a tab 78 for a "Start Menu Programs" property sheet. The window includes an OK button 84 accepting user-made property changes, a Cancel button 82 for not accepting user-made property changes and an Apply button 84 for immediately applying changes that have been made while keeping the frame window 74 open.

The "Taskbar Options" property sheet 76 includes a preview area 86 that illustrates the effect of selecting or not selecting check boxes 88, 90 and 92. Figure 15A shows an example wherein the "Always on top" check box 88 has been checked so that the taskbar is always displayed on top of other components in the user interface. The preview area 86 of the property sheet 76 shows that the illustrated portion of the taskbar 94 is positioned to obscure the window 96. This appearance is in contrast to the appearance of the taskbar in the preview area 86 of Figure 14, where the taskbar is obscured by the window.

Figure 15B illustrates an example wherein the "Auto hide" check box 90 has been selected so as to place the taskbar in auto hide mode. As the preview section 86 of the property sheet 76 indicates, in auto hide mode, the taskbar is not readily visible. Only a small sliver 98 of the taskbar is visible. The auto hide option is provided

to allow users to exploit more of the available screen space on the video display 18 when needed.

The auto hide mode hides the taskbar. The taskbar 28 is hidden when a user opens a window on the desktop. When no more opened windows remain, the task bar reappears. In addition, a user may cause the taskbar to reappear when a window is opened by positioning the mouse cursor to point along the edge of the video display to which a taskbar is anchored. The taskbar disappears when the user moves the cursor out of this area. Programmatically, the auto hide mode is instituted by the taskbar monitoring where the windows are opened and determining when to display the bar and when to hide the task bar.

Figure 15C shows an example wherein the "Show small icons in Start menu" list box 92 has been selected to show small icons in the start menu rather than large icons. The system provides the icons in two varieties: small and large. Small icons, as the name implies, differ from the large icons in that they are smaller in dimension. When check box 92 is checked, the preview area 86 of property sheet 76 shows small icons in the start menu 100.

The start menu button 32 serves as the vehicle for opening the start menu. The start menu acts as a centralized place for a user to gain access to programs, documents, system settings, help information and the like. A user causes the start menu 101 (Figure 16A) to be displayed by positioning the mouse cursor to point at the start menu button 32 and then clicking the left mouse button. The start menu 101, by default, includes a number of menu items. The "Find" menu item 104 allows a user to invoke a querying dialog to locate programs, documents or other entities within the system. The "Help" menu item 106 invokes a menu dialog that allows a user to find useful help information about various topics. Thus, if a

user needs help or needs to find something, the user needs only to access the start menu 12 and choose one of these menu items.

The "Settings" menu item 108 allows a user to adjust system settings. The "Documents" menu 110 allows a user to gain access to up to fifteen of the most recently used documents. Figure 16B shows an example where cascade menu 103 is shown displaying three documents when the "Documents" menu item is selected. The system 16 maintains a cache of the 15 most recently opened documents and icons for these documents. A menu item for the documents on this list are displayed when the "Documents" menu item 110 is selected.

The "Programs" menu item 112 allows a user to gain access to the programs menu 105 (Figure 16C) that cascades off the start menu 101. The programs menu 105 displays programs and program groups that a user may select. The "Run" menu item 114 enables a user to select a program to run. Lastly, the "Shut Down" menu item 116 displays a shut down dialog to allow a user to shut down the system. The arrows 118 shown for menu items 104, 108, 110, and 112 indicate that these menu items have cascading menus that are displayed when the mouse cursor is positioned to point at these items.

Figure 17 shows an example of a cascading menu 120 for the "Settings" menu item 108 in the start menu 101. When the mouse cursor is positioned to point at one of the menu items in the start menu 101, the item is highlighted. In Figure 17, the "Settings" menu item 108 is highlighted, and the corresponding cascading menu 120 is displayed to show the user the available setting options. These settings options include an option 122 for opening a control panel window. These options are highlighted as the mouse cursor is positioned to point at them. In the example shown in Figure 18, the control panel option 122 is shown highlighted. The user may then

open the control panel window 124 (as shown in Figure 19) by single clicking the left mouse button while the cursor points to the control panel option 122.

The above described menu items shown in the start menu 101 are displayed by default. A user may configure the start menu to add additional items and to remove certain items. The system provides protections that do not allow a user to remove certain items from the start menu 101. The user may configure the start menu by invoking the "Start Menu Programs" property sheet through the taskbar context menu 68. The "Start Menu Programs" property sheet 128 is displayed in the frame window 74 that was described above relative to the property sheet for the taskbar. The "Start Menu Programs" property sheet 128 includes an add button 130 for adding items to the start menu. A remove button 132 is provided to remove items from the start menu and an advanced button 134 is provided for performing advanced customization of the start menu.

When the user presses the add button 130, a dialog 138 that is part of a sequence for adding an item to the start menu is displayed as shown in Figure 21. The dialog 138 includes an edit box 140 for entering the name of the item to be added. Alternatively, a user may activate a browse button 142 which invokes additional dialogs to enable a user to browse through items to be added. The dialog 138 includes a next button 146 for progressing to a next dialog in the sequence and a back button 144 for returning to the previous dialog in the sequence. Lastly, a cancel button 148 is provided to cancel the addition of an item to the start menu.

When the user depresses the browse button 142 (Figure 21) a browse dialog 150 (Figure 22) is displayed. The browse dialog 150 includes a window 154 which shows folders and items that may be added to the start menu 101. A user may select one of the items displayed within window

154 or may type a file name in edit box 152. The user then depresses the open button 156 to request the selected item to be added to the start menu 101. The user may cancel the browse dialog 150 by hitting the cancel button 158. A drop down list box 160 is provided in the browse dialog 150 to present options regarding the places where the user may look for items to add to the start menu. An additional drop down list box 162 is provided for a user to select the type of file that is to be displayed in window 154. Figure 23 shows an example wherein a user has chosen the windows folder and selected the "Calc" program to be added to the start menu. In particular, the user has clicked on the icon associated with "Calc." As a result, the name of this file appears in the edit box 152.

When the user presses the open button 156 in the browse dialog 150, a "Select Program Group" dialog 160 is shown. This dialog is used to select to program group to which the item is to be added to within the cascading menus that are accessible from the start menu 101. In the example shown in Figure 24, a user has selected the start menu 101 as the program group by positioning the mouse cursor to point at the entry for the start menu that is shown in scroll box 168 and clicking the left mouse button. The user also has the option of pressing a new folder button 166 to define a new folder that is to be added to the cascading start menus. When the user depresses the next button 146, the selected item is added to the start menu 101.

With regard to Figure 20, if a user selects the remove button 130, a "Remove Shortcuts/Folders" dialog 170 (Figure 25) is displayed. This dialog 170 displays a tree view 172 of the items that are currently accessible via the start menu 101. To remove an item from the start menu, a user selects an item in the tree hierarchical view 172 and presses the delete button 174. Selected items have their labels highlighted. In the example shown in

Figure 25, the "Calc" program is selected. When the user wishes to exit dialog 170, the user selects the close button 176.

When the user selects the advanced button 134 (Figure 20), a window 180 (Figure 26) for exploring the start menu is provided. As the items shown in the explorer window 180 support OLE and WINDOWS drag and drop operations, they may be dragged and dropped to manipulate the start menu.

When a user is completed with the property sheets of the frame window 74 (Figure 20), the user may press the OK button 80 to accept the changes made by the user or press the cancel button 82 to not accept the changes made by the user. For the above-described examples, suppose that the user has decided to add the "Calc" item to the start menu via changes made to the property sheets. The start menu 101 then appears as shown in Figure 27 with an additional menu item 182 for the "Calc" program. A user may then directly access the "Calc" program from the initial start menu 101, rather than traversing the cascaded menus to get access to the program.

The above-described examples have generally related to instances wherein the taskbar 28 is positioned at the bottom of the user interface. It should be appreciated that the start menu 101 may be displayed at different locations and cascades differently depending upon where the taskbar 28 is anchored. Figure 28 shows an example of the start menu 101 when the taskbar 28 is anchored to the right side of the user interface. The start menu 101 is under control of the taskbar process in the operating system 26. As a result, the taskbar determines where to draw the start menu. The location at which to draw the start menu changes when the taskbar 28 is moved to be anchored at a new location. In general,

the start menu 101 is displayed to the interior side of the taskbar 28.

The clock 34 on the taskbar 28 has associated property sheets. These property sheets may be accessed by either double clicking on the clock 34 or by invoking context menu 188 (Figure 29). The context menu 188 is displayed in response to a user positioning a mouse cursor to point at the clock 34 and pressing the right mouse button. This context menu 188 includes an "Adjust Date/Time" menu item 190. The property sheets for the clock 34 include a "Date & Time" property sheet 190 (as shown in Figure 30) and a "Time Zone" property sheet 192 (as shown in Figure 31). These property sheets enable a user to select a current month, year, day, hour, minute, second and time zone.

As was discussed above, the taskbar 28 is designed to be consistently visible such that it is not readily obscured by opened windows. As shown in Figure 32, when a window for a word processing program is maximized, the maximized word processing window 200 does not obscure the taskbar 28. This is achieved by setting the working area available to application window to exclude the taskbar region. The taskbar does not cover any part of the word processing window, rather their borders merely meet.

One complication that arises with the taskbar concerns an instance where a user wishes to drag an object to an active window that is currently minimized. In the preferred embodiment of the present invention, the user may drag the object to the taskbar button that is associated with the application. In particular, as shown in Figure 33, initially the object is dragged using the mouse 14 to be positioned over one of the taskbar buttons (step 202 in Figure 33). The object must remain positioned with the mouse button continually pressed over the taskbar button for a predetermined period of time

(step 204). When the predetermined period of time has elapsed, the application window associated with the taskbar button is made visible in a non-minimized state (step 206) and the user may drop the object in the application window (step 208).

The present invention has been described with reference to a preferred embodiment thereof. Those skilled in the art will appreciate the various changes in form and detail that may be made without departing from the intended scope of the invention as defined in the appended claims.

4. Brief Description of Drawings

Figure 1 is a block diagram of the computer system that is suitable for practicing the preferred embodiment of the present invention.

Figure 2 is an example of a user interface that employs the taskbar of the preferred embodiment of the present invention.

Figure 3 illustrates the appearance of the taskbar of Figure 2 when two application programs have been opened and their windows have been minimized.

Figure 4 shows an example of how a program may be made fully visible from taskbar buttons on the taskbar.

Figure 5 shows an example of how taskbar buttons may be automatically resized based on the available amount of space on the taskbar.

Figure 6 is a flowchart illustrating the steps that are performed to resize the taskbar.

Figure 7 is a diagram illustrating an example of the appearance of the taskbar when it is being resized.

Figure 8 shows an example of the appearance of a taskbar of a different size than the taskbar of Figure 7.

Figure 9 is a flowchart illustrating the steps that are performed to reanchor the taskbar to a new edge of the video display.

Figure 10A illustrates the taskbar when it is anchored to the right edge of the video display.

Figure 10B illustrates the appearance of the taskbar when it is anchored to the top edge of the video display.

Figure 10C illustrates the appearance of the taskbar when it is anchored to the left edge of the video display.

Figure 11 illustrates an example context menu for a taskbar button.

Figure 12 illustrates an example context menu for the taskbar.

Figure 13 illustrates an example context menu for the taskbar when a window tiling option has been chosen.

Figure 14 illustrates the "Taskbar Options" property sheet for the taskbar.

Figure 15A illustrates the effect of choosing the "Always on top" check box in the property sheet of Figure 14.

Figure 15B illustrates the effect of selecting the "Auto hide" check box in the property sheet of Figure 14.

Figure 15C illustrates the effect of choosing the "Show small icons in Start menu" check box in the property sheet of Figure 14.

Figure 16A illustrates an example of a start menu in accordance with the preferred embodiment of the present invention.

Figure 16B illustrates an example of a cascading documents menu that may be accessed via the start menu.

Figure 16C illustrates a cascading programs menu that may be accessed via the start menu.

Figure 17 illustrates a cascading settings menu that may be accessed via the start menu.

Figure 18 illustrates an instance wherein a "Control Panel" option is selected in the cascading settings menu of Figure 17.

Figure 19 illustrates an open control panel window that is accessed via the cascading settings window of Figure 18.

Figure 20 illustrates a "Start Menu Programs" property sheet for the start menu.

Figure 21 shows a dialog that may be accessed via an "Add" button on the property sheet of Figure 20.

Figure 22 illustrates a browse dialog for adding an item to the start menu.

Figure 23 shows the browse dialog of Figure 22 when used to select an item to be added to the start menu.

Figure 24 shows another dialog in a sequence of dialogs for adding an item to the start menu.

Figure 25 shows a dialog for removing items from the start menu.

Figure 26 shows an open explorer window for performing advanced manipulation of the start menu.

Figure 27 shows an example of a start menu that has been configured to add an item to it.

Figure 28 illustrates the cascading of the start menu when the task bar is anchored to the right edge of the video display.

Figure 29 illustrates a context menu that may be generated for the clock on the task bar.

Figure 30 illustrates an example "Date & Time" property sheet for the clock on the taskbar.

Figure 31 illustrates a "Time Zone" property sheet for the clock on the taskbar.

Figure 32 illustrates how an opened word processing application window does not obscure the taskbar.

Figure 33 is a flowchart illustrating steps that are performed to drag an object to a taskbar button.

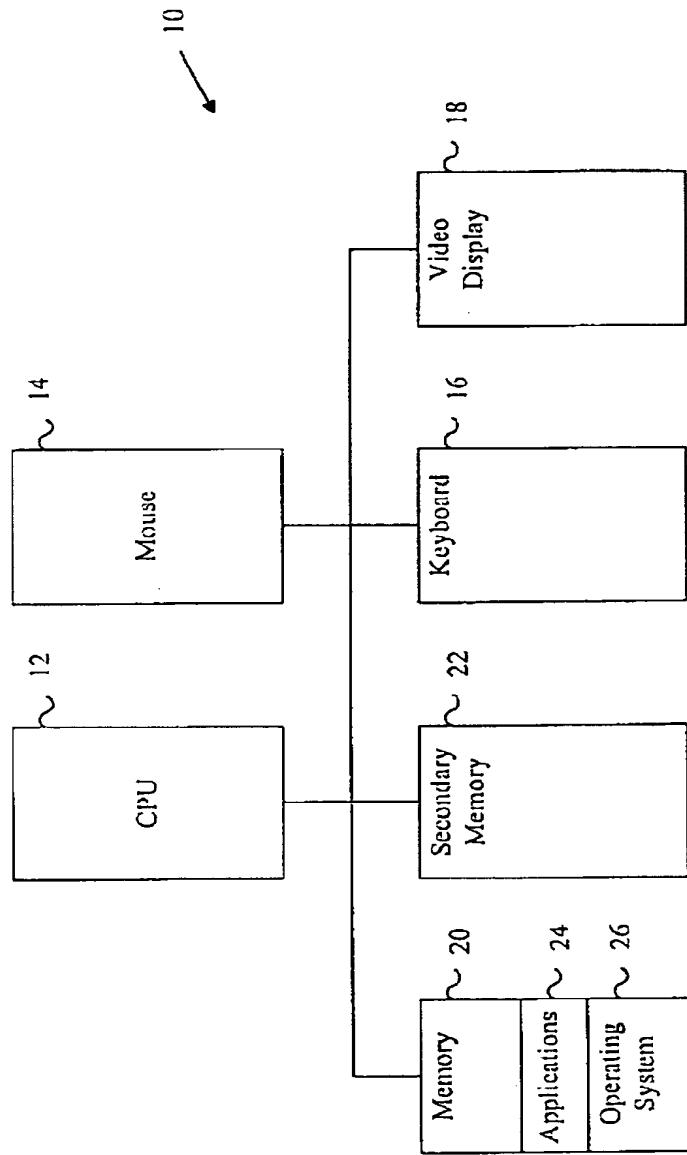


FIG. 1

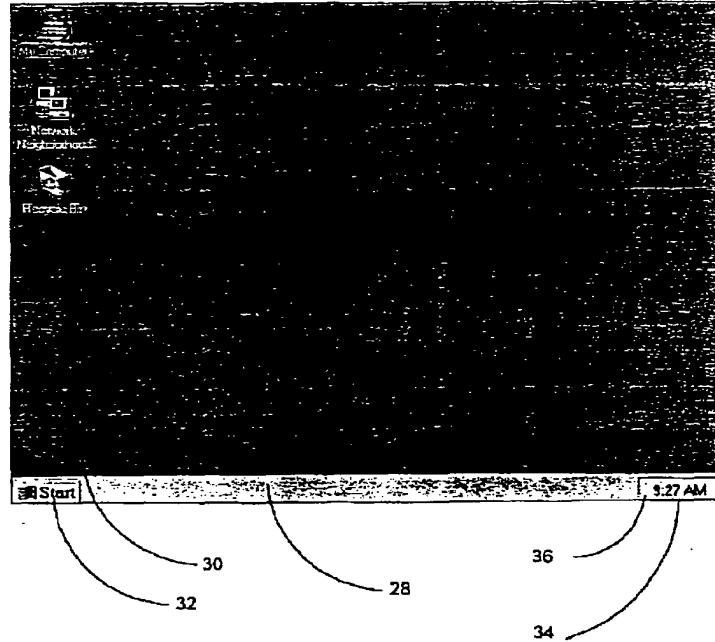


FIG. 2

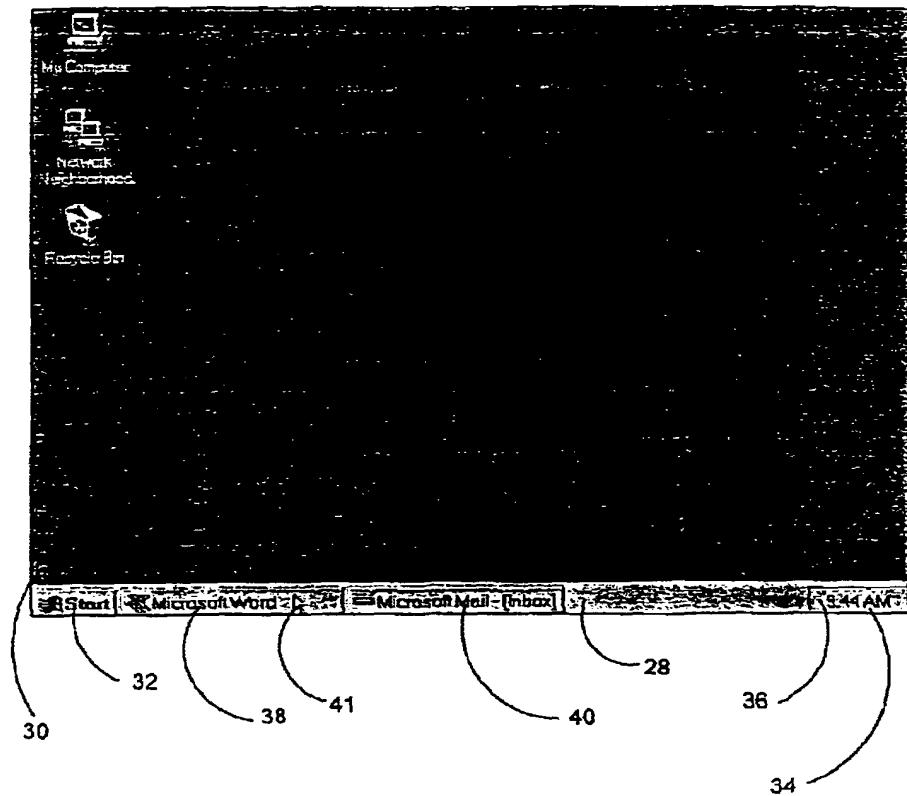


FIG. 3

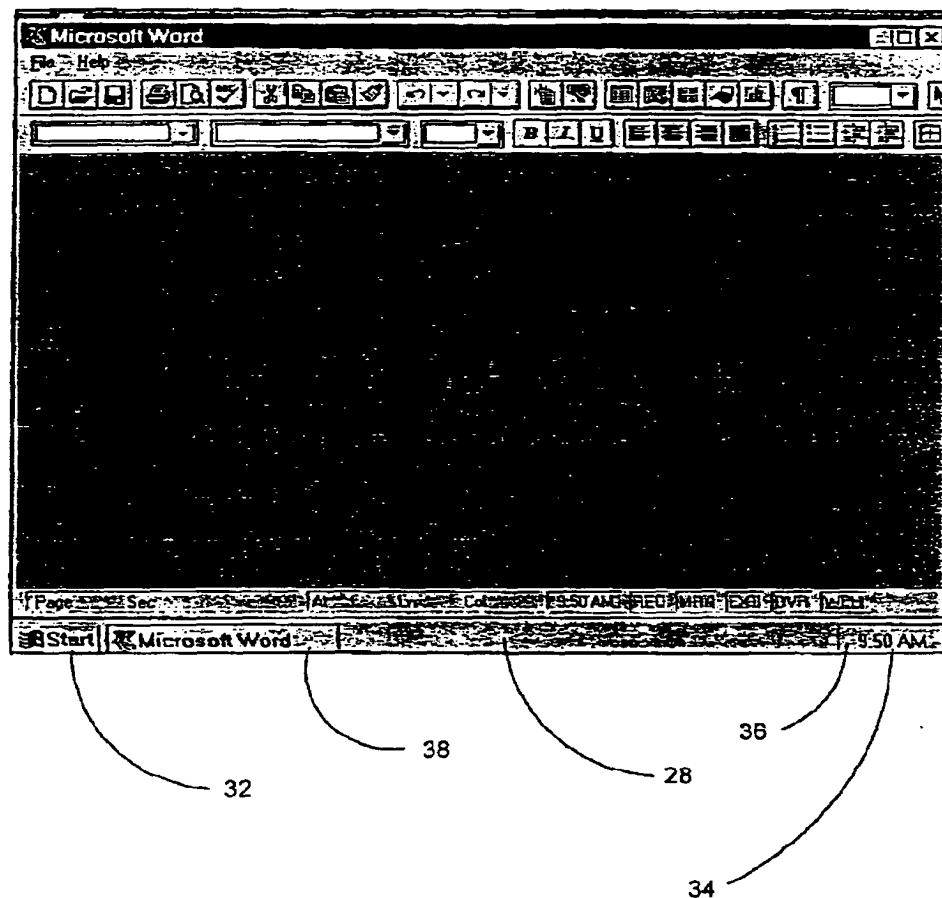


FIG. 4

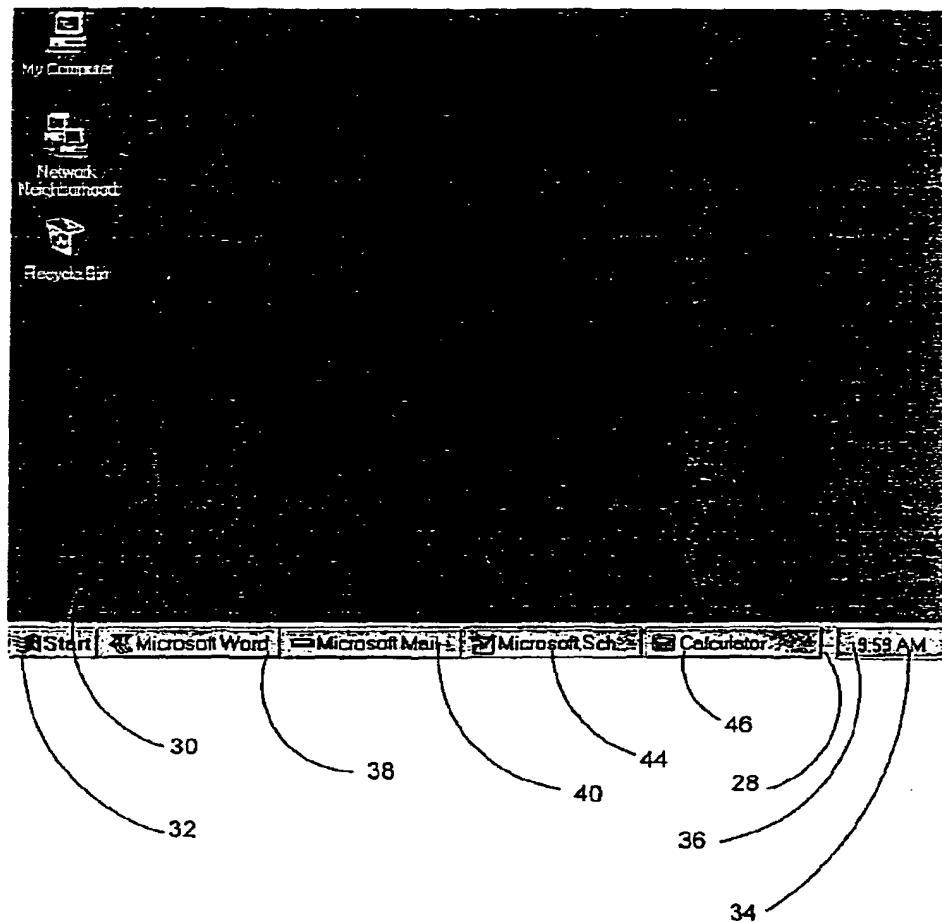


FIG. 5

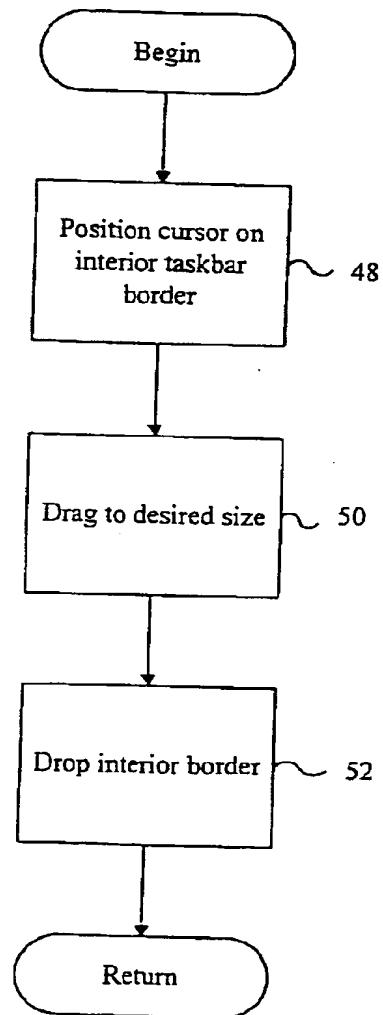


FIG. 6

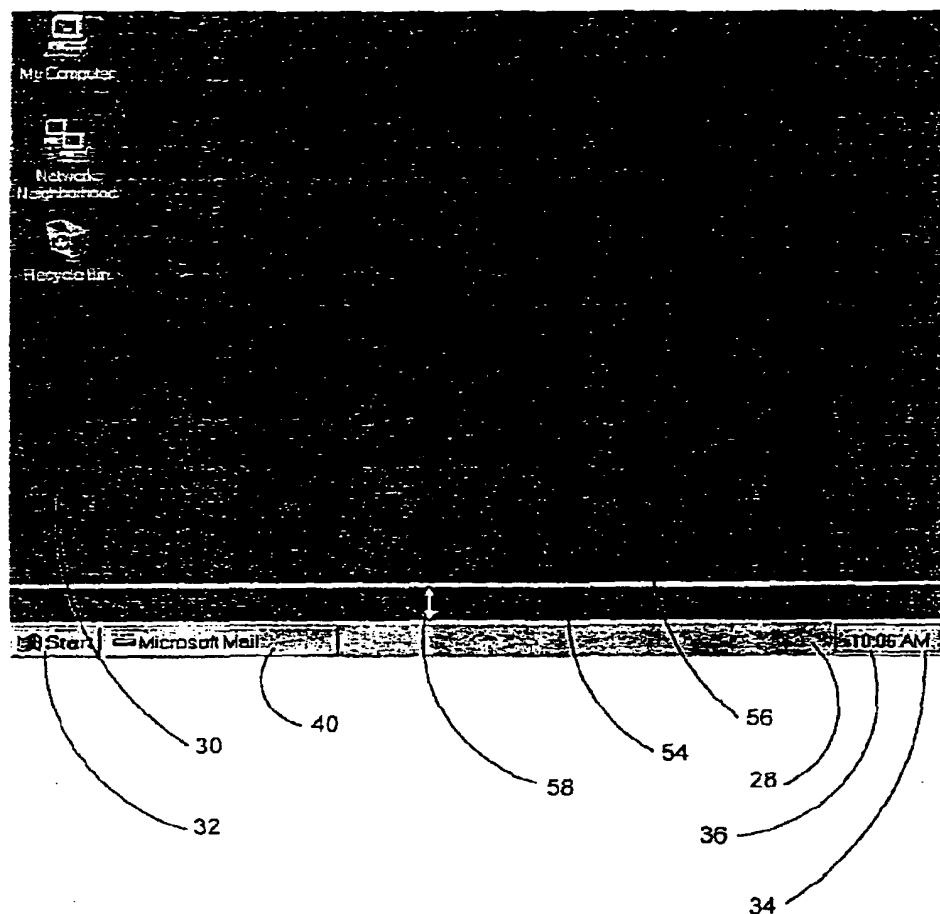


FIG. 7

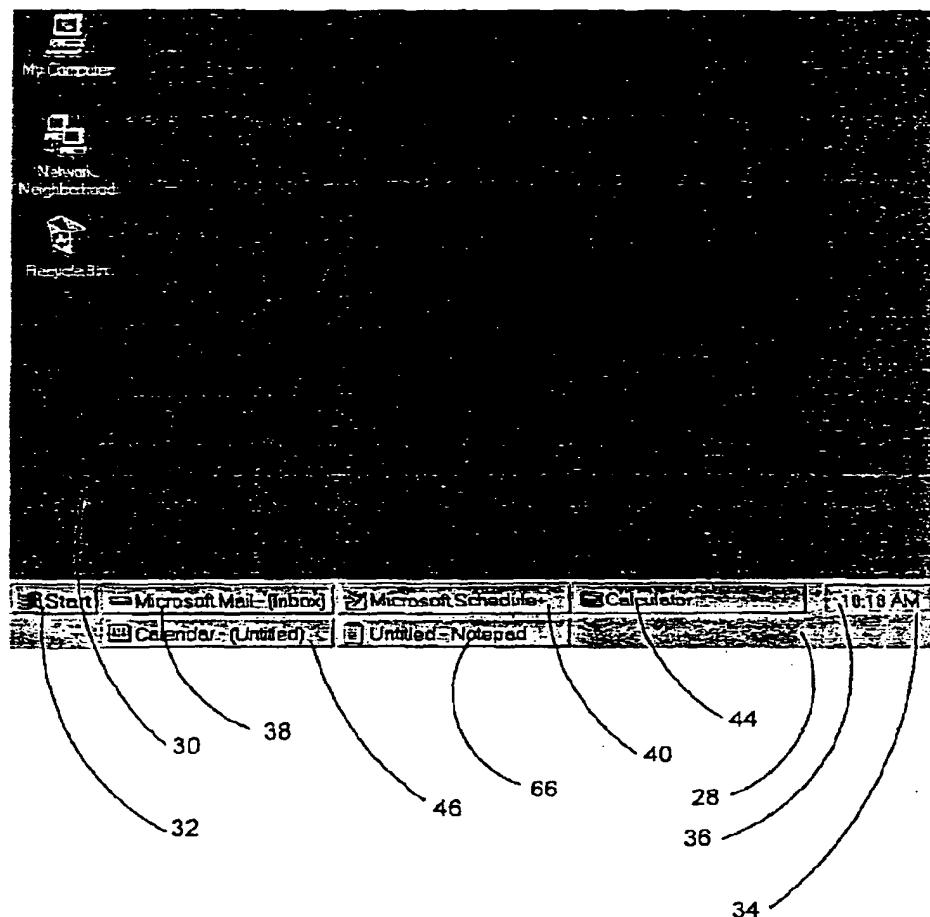


FIG. 8

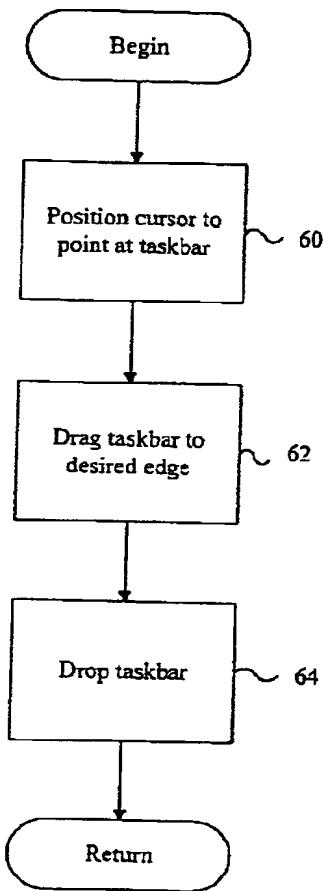


FIG. 9

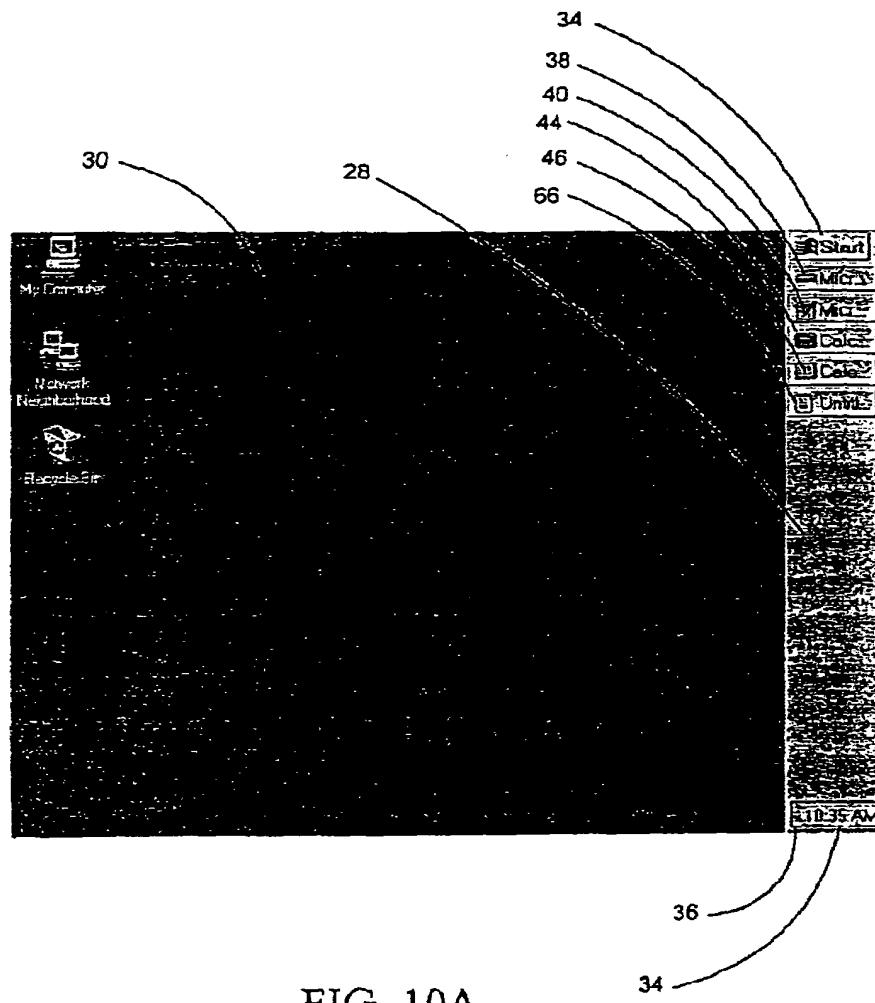


FIG. 10A

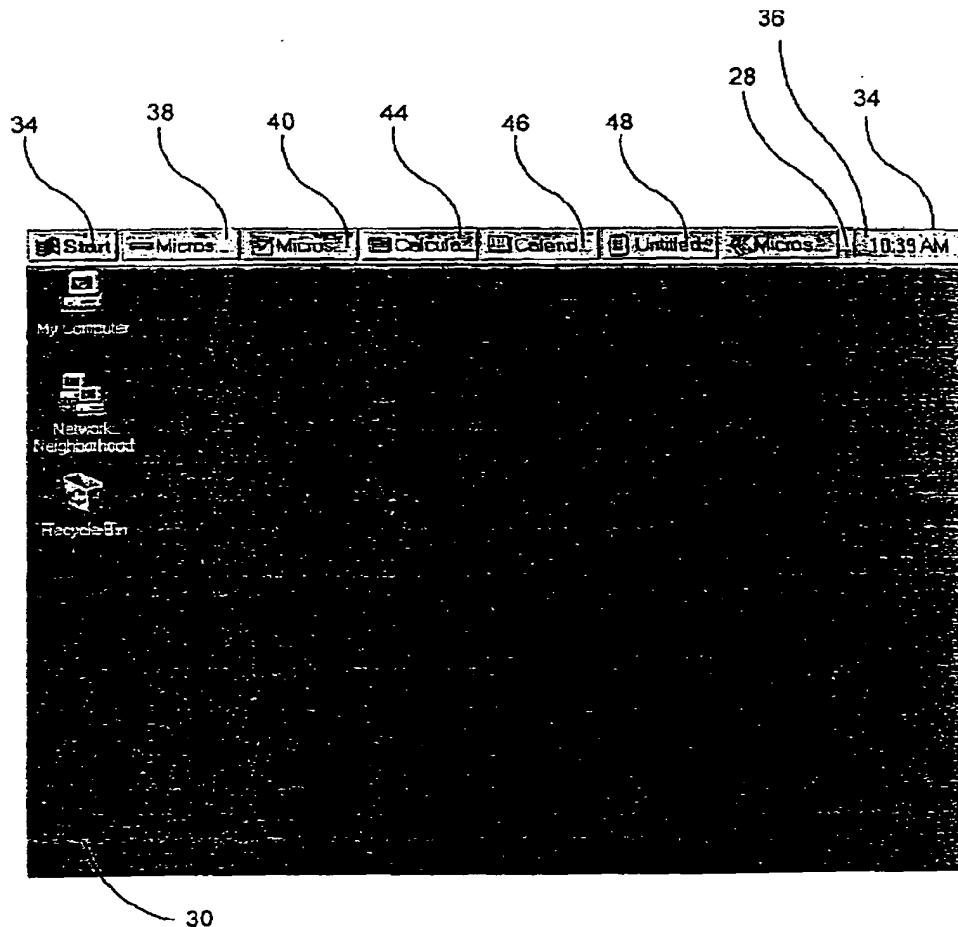


FIG. 10B

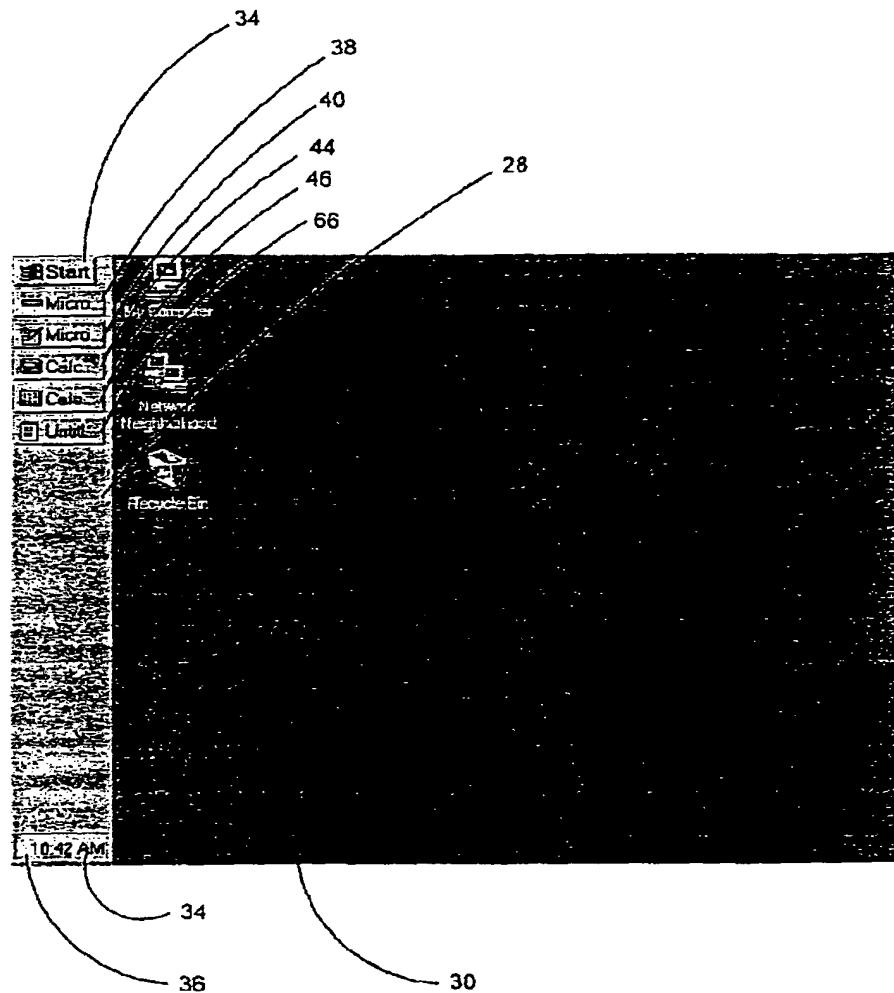


FIG. 10C

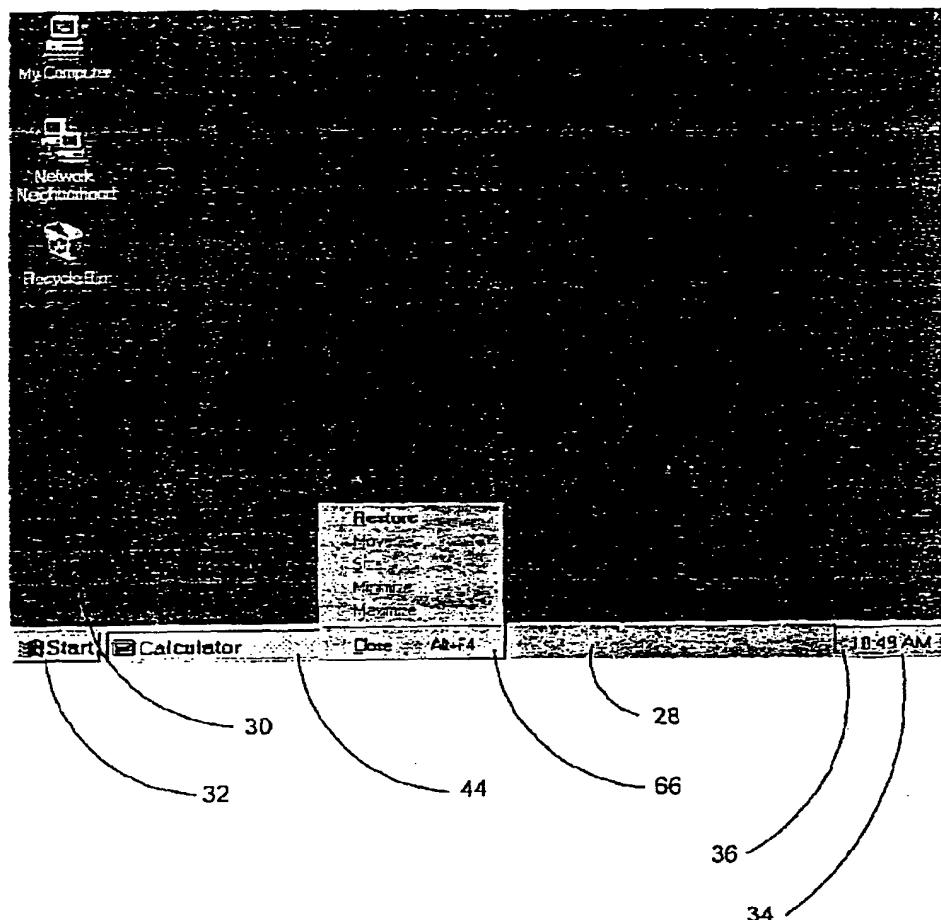


FIG. 11

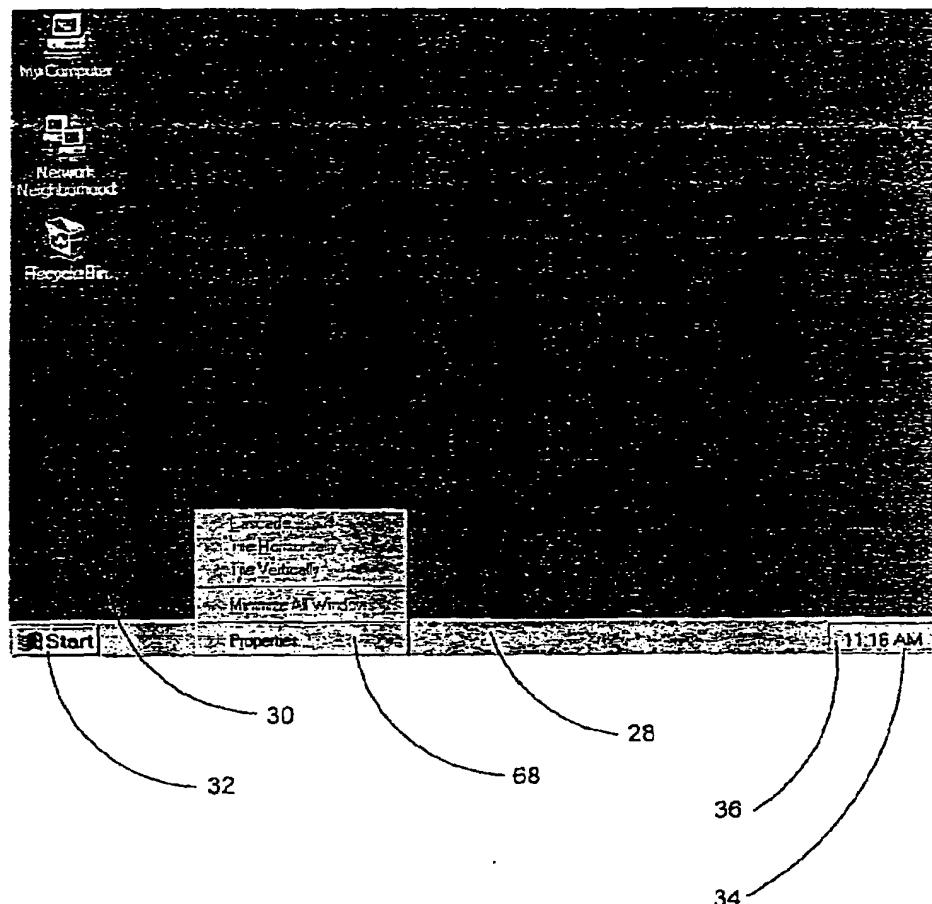


FIG. 12

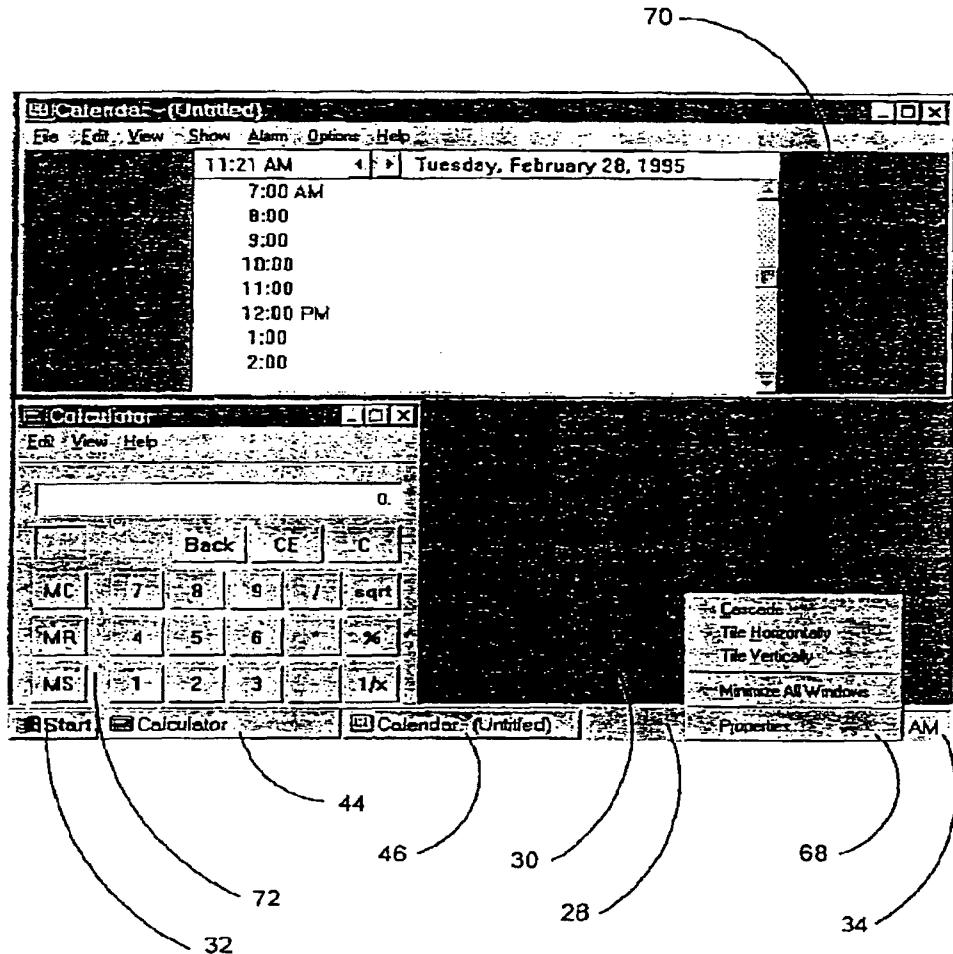


FIG. 13

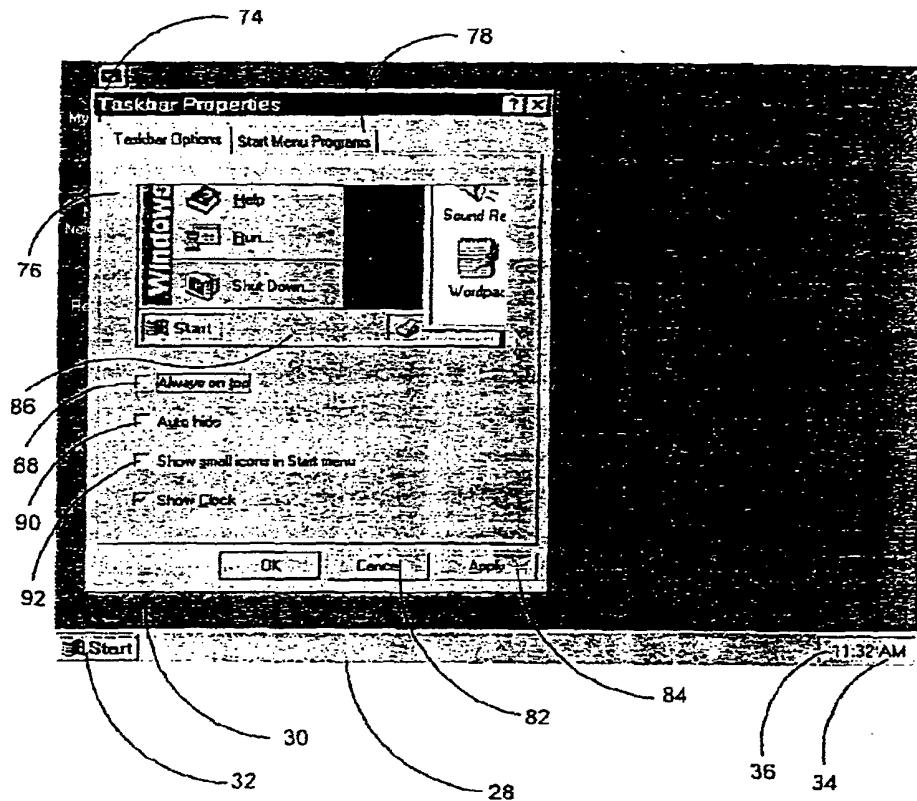


FIG. 14

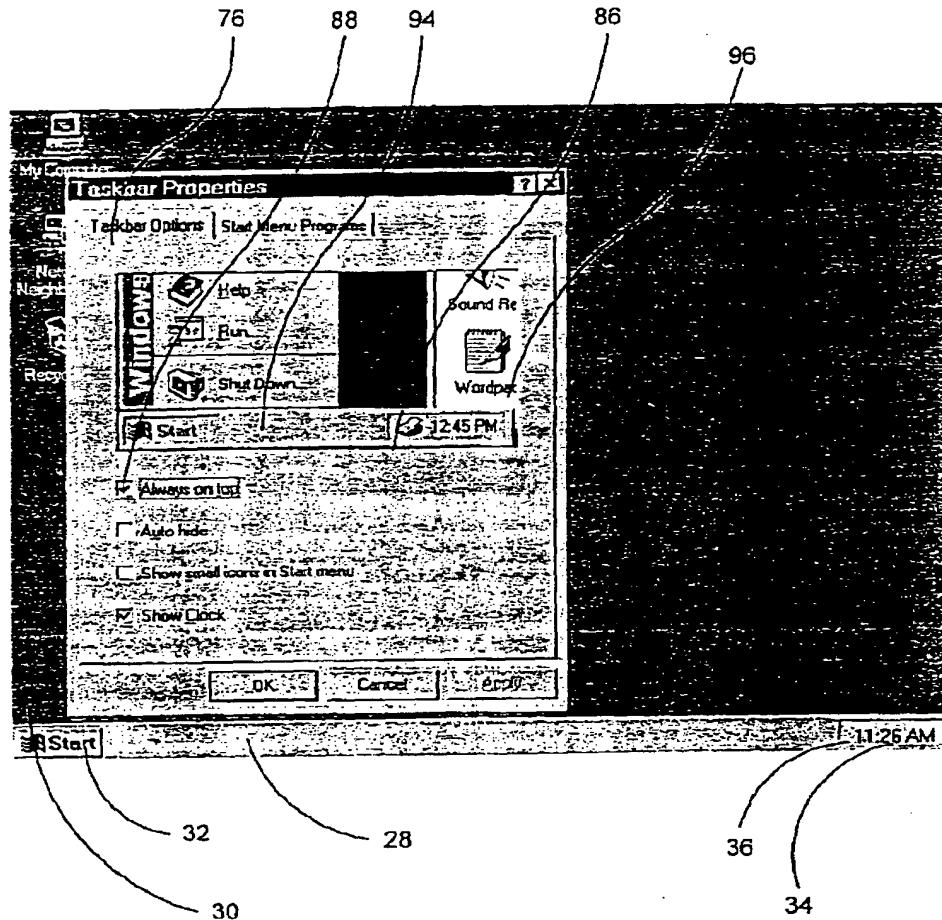


FIG. 15A

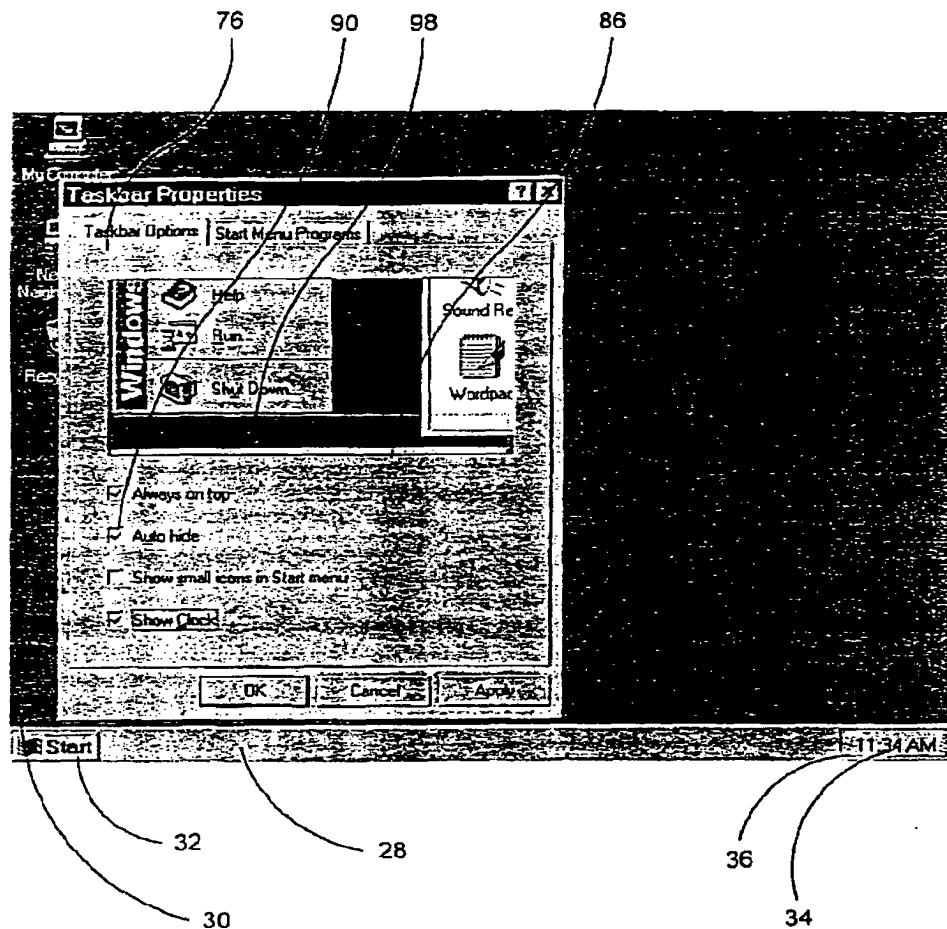


FIG. 15B

(100)

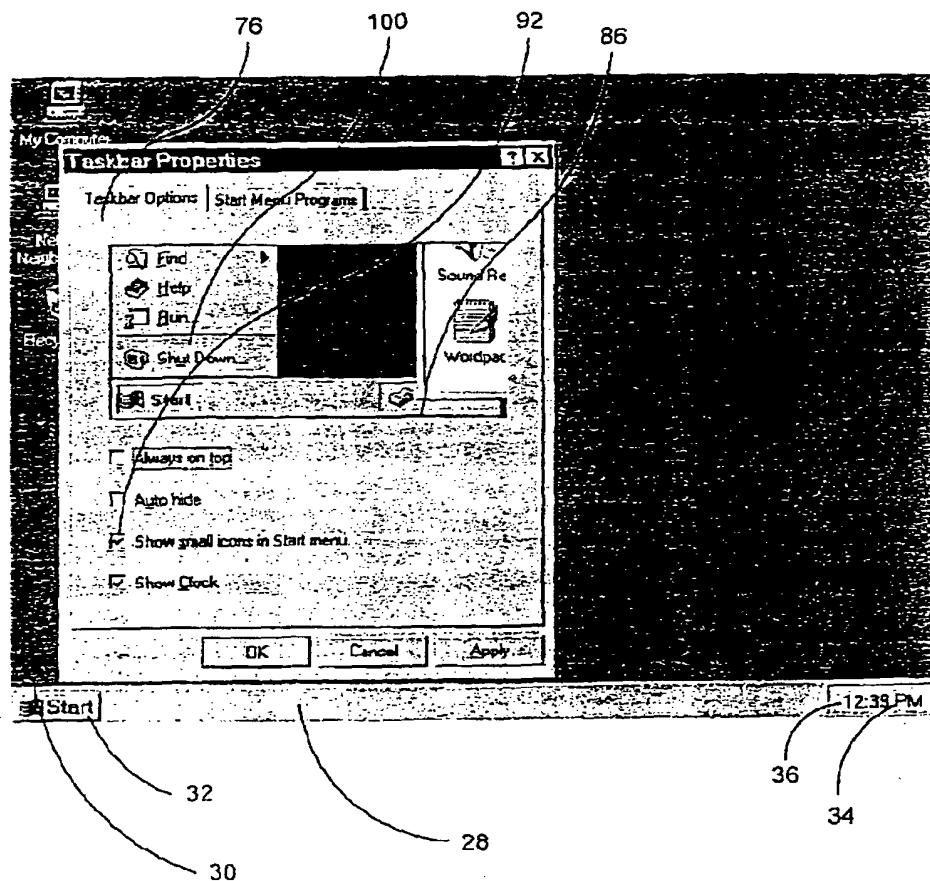


FIG. 15C

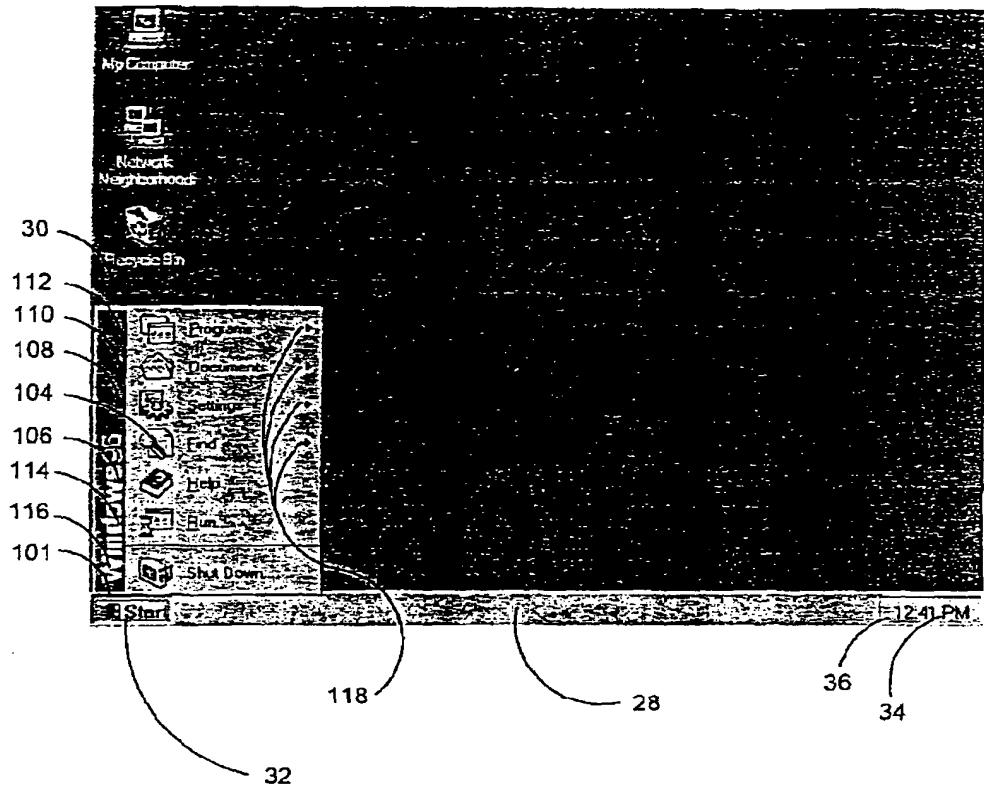


FIG. 16A

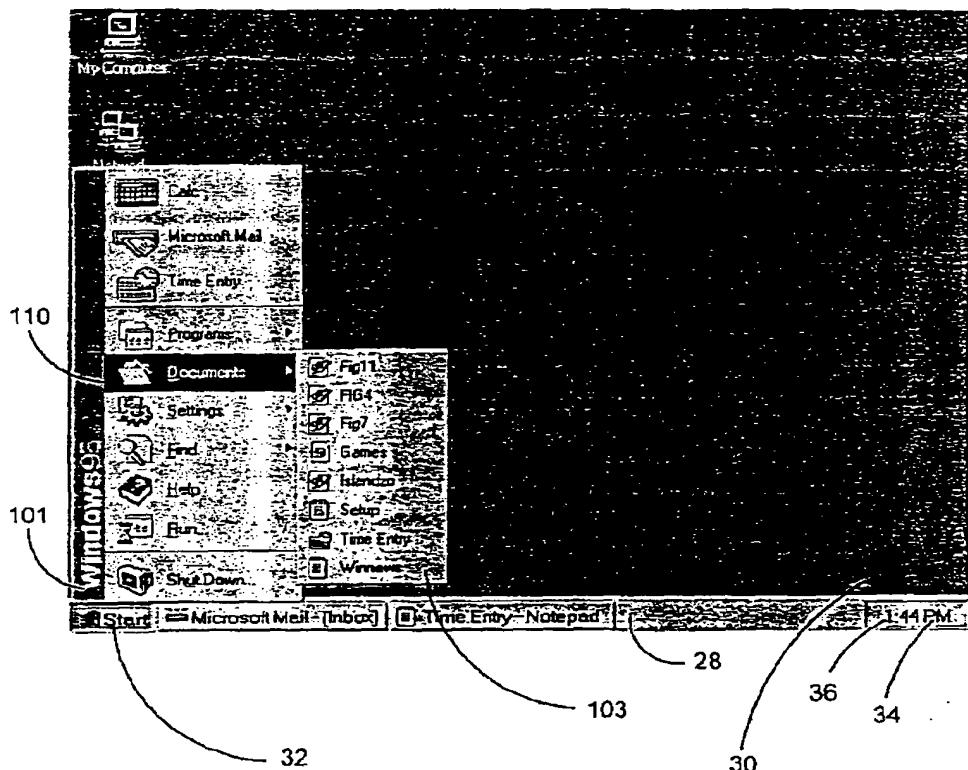


FIG. 16B

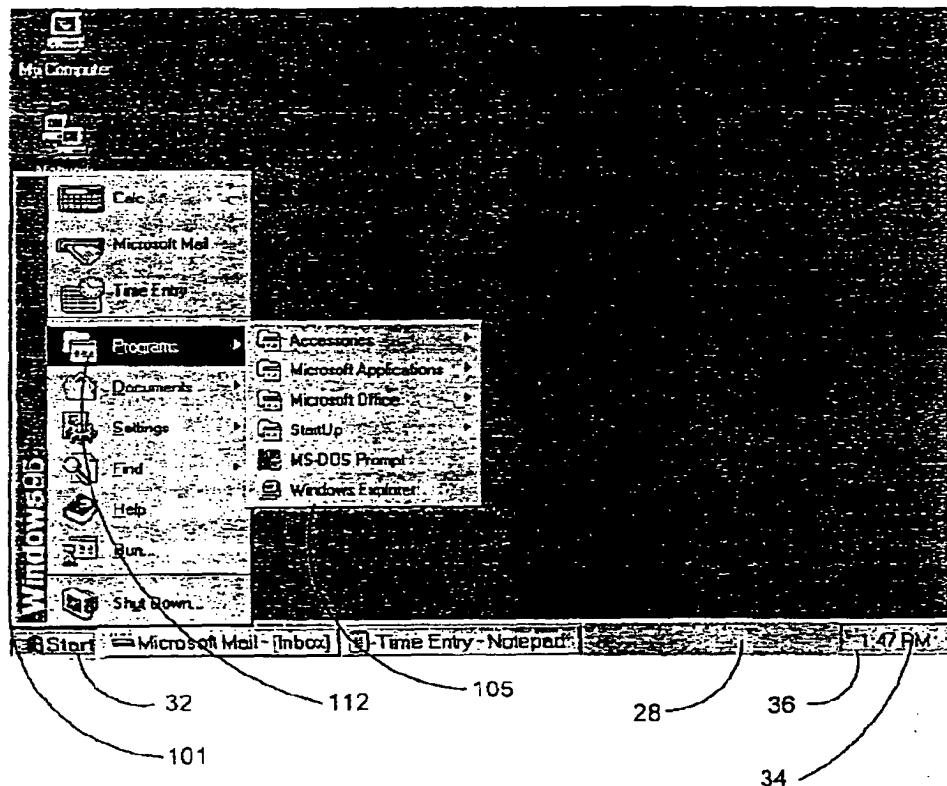


FIG. 16C

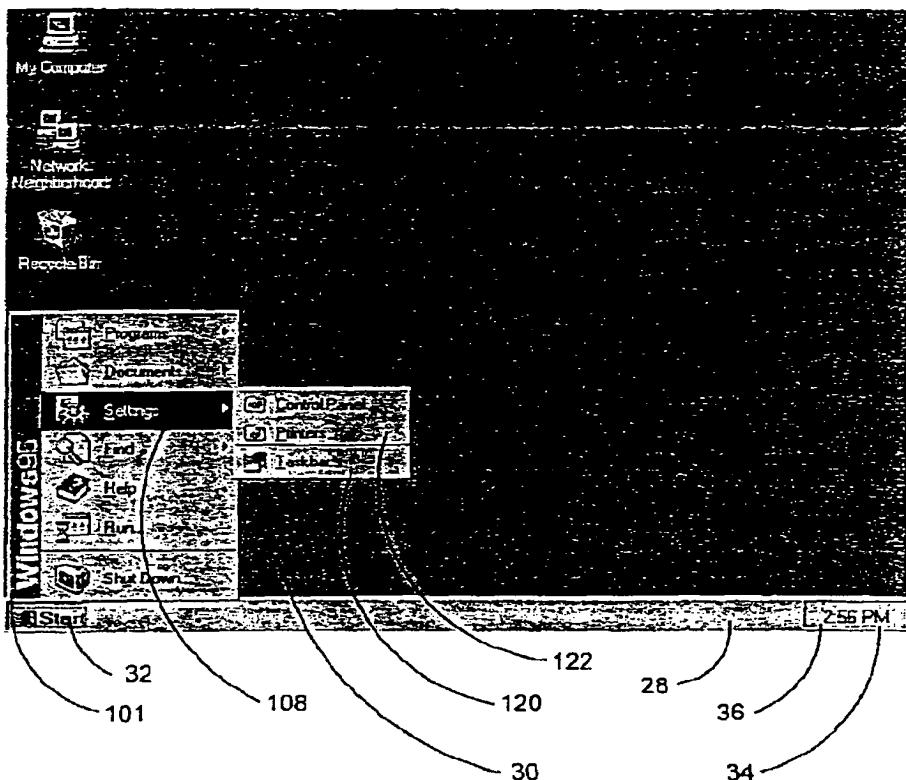


FIG. 17

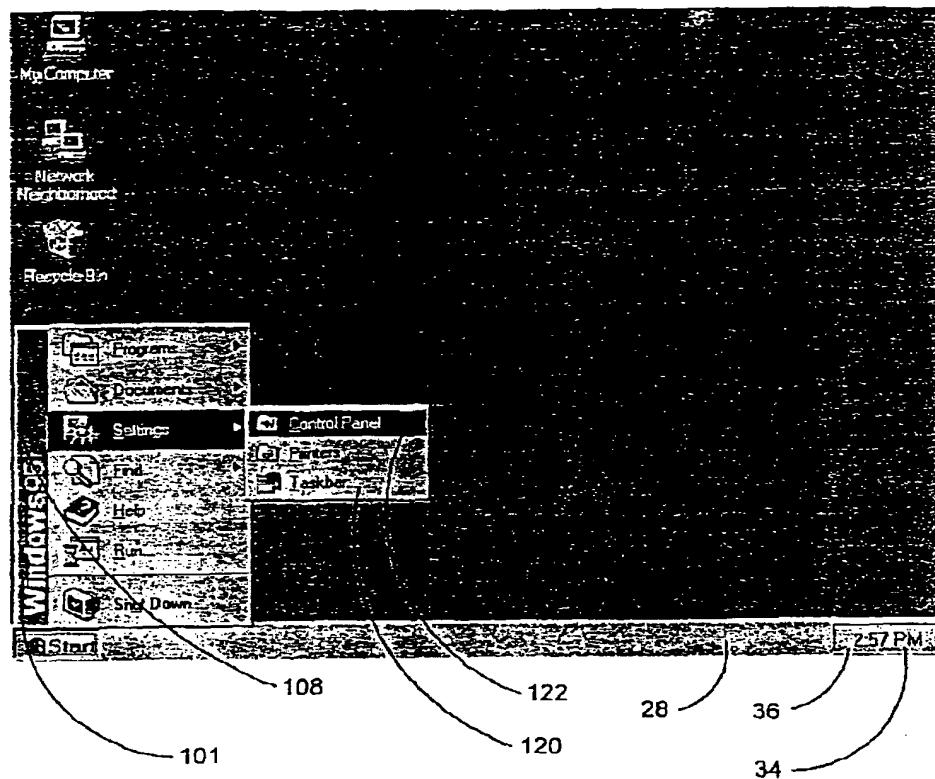


FIG. 18

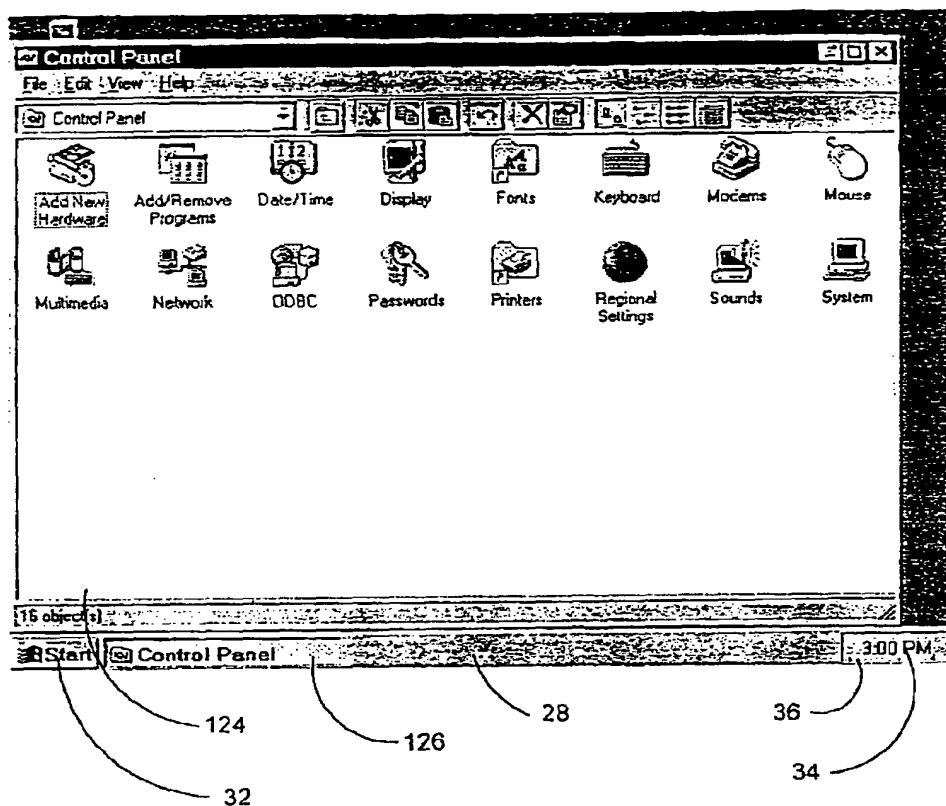


FIG. 19

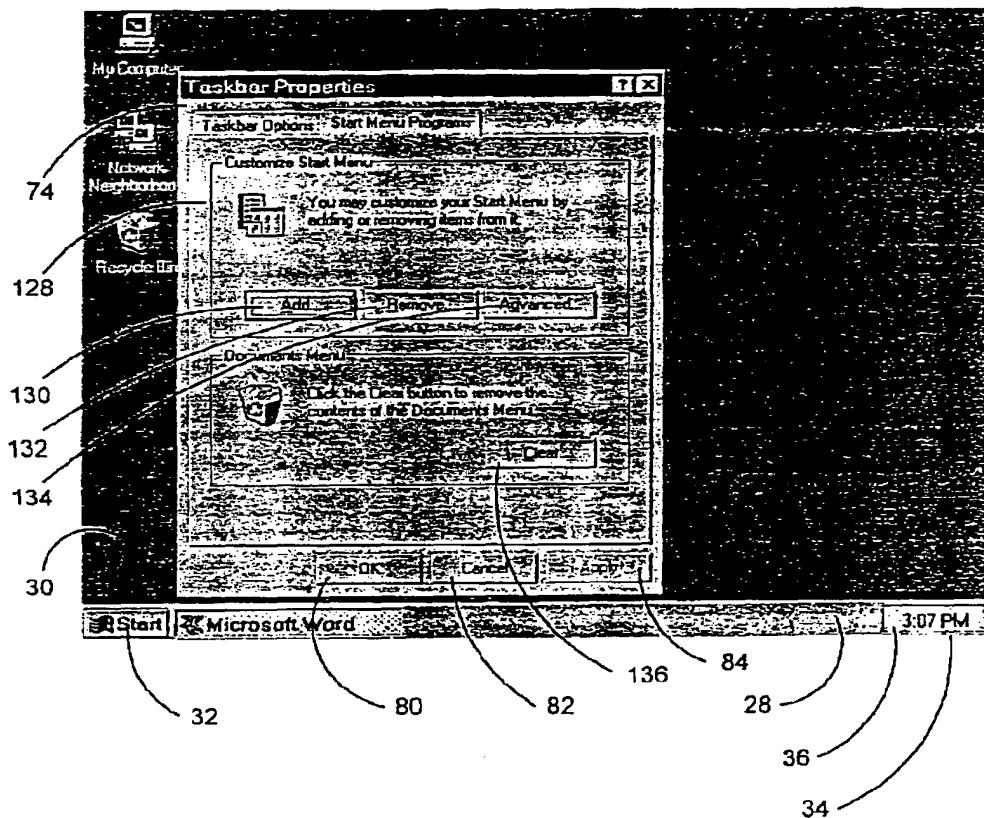


FIG. 20

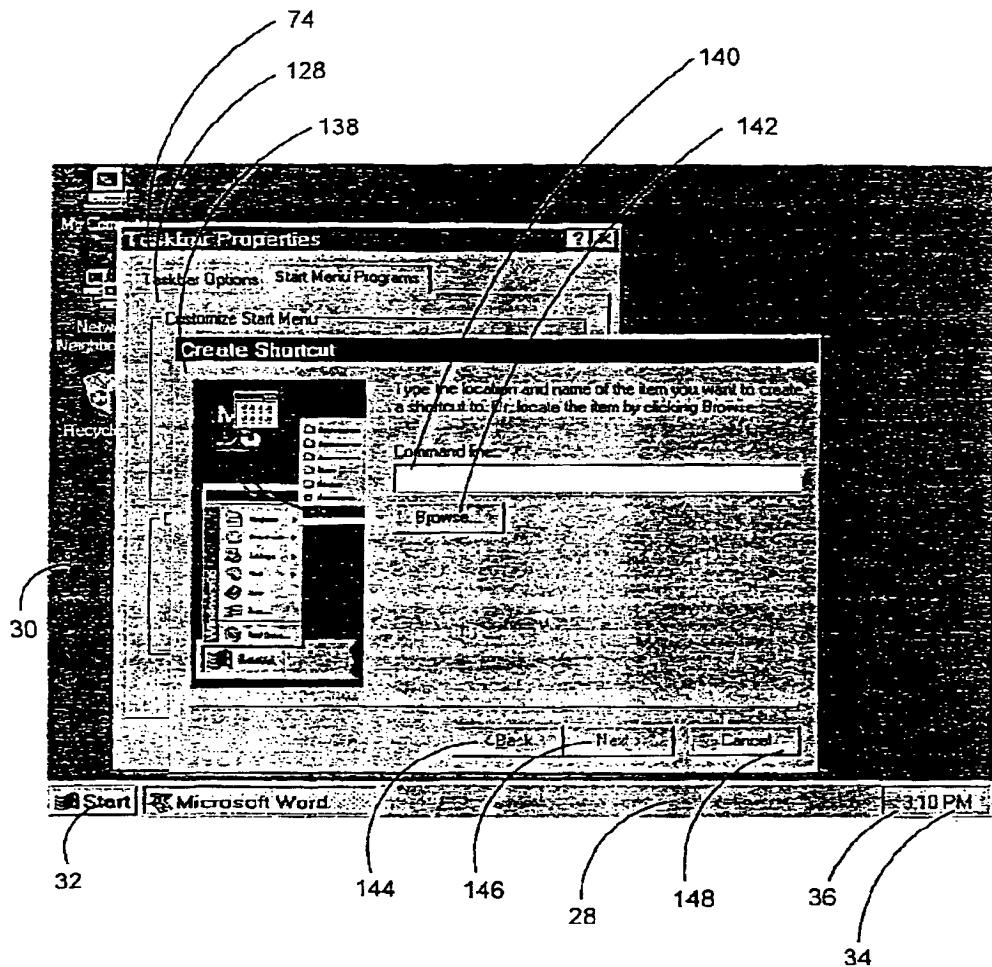


FIG. 21

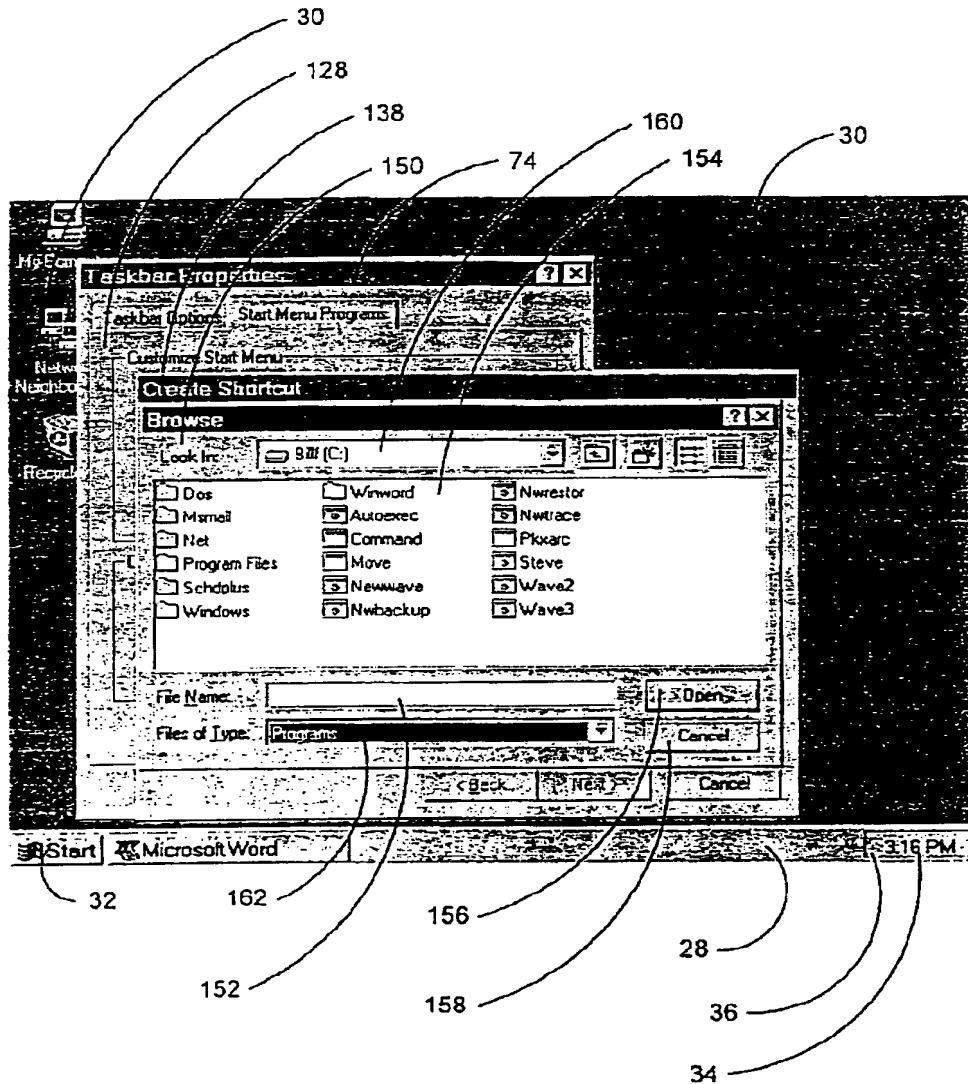


FIG. 22

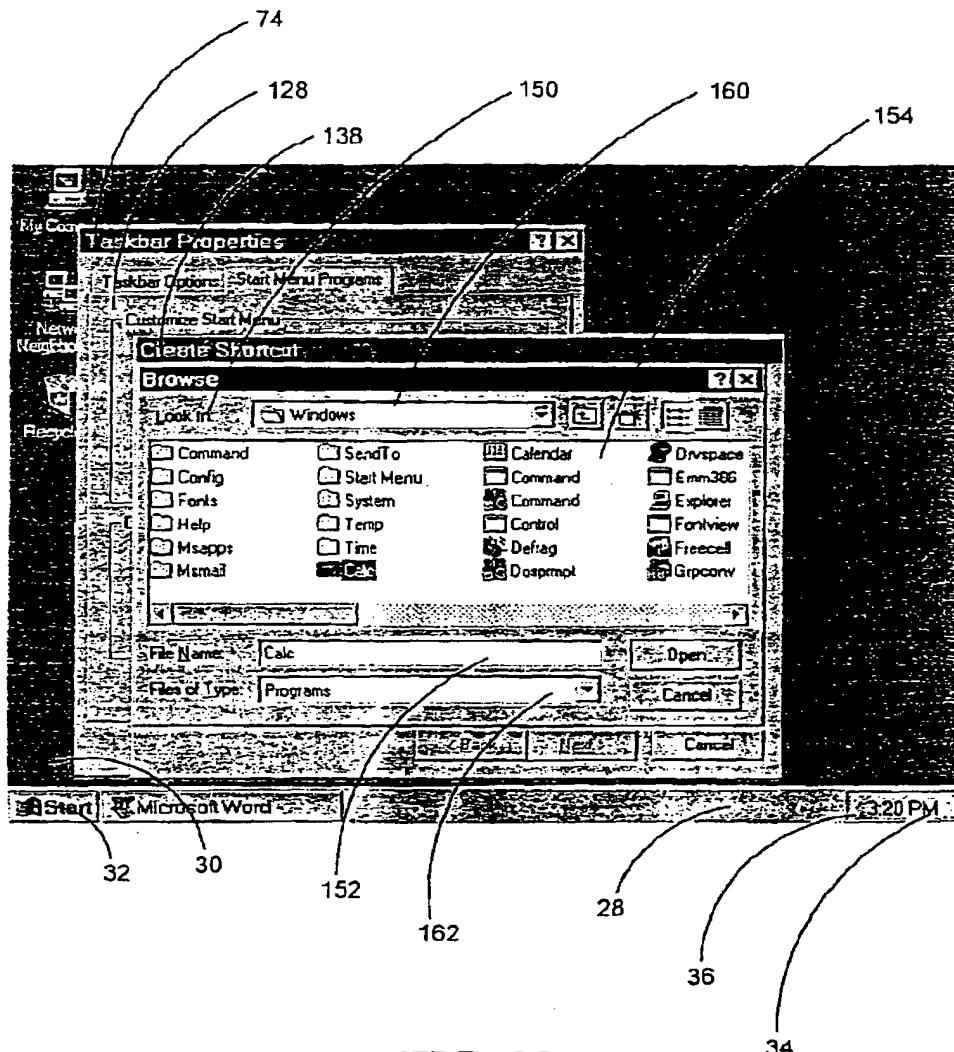


FIG. 23

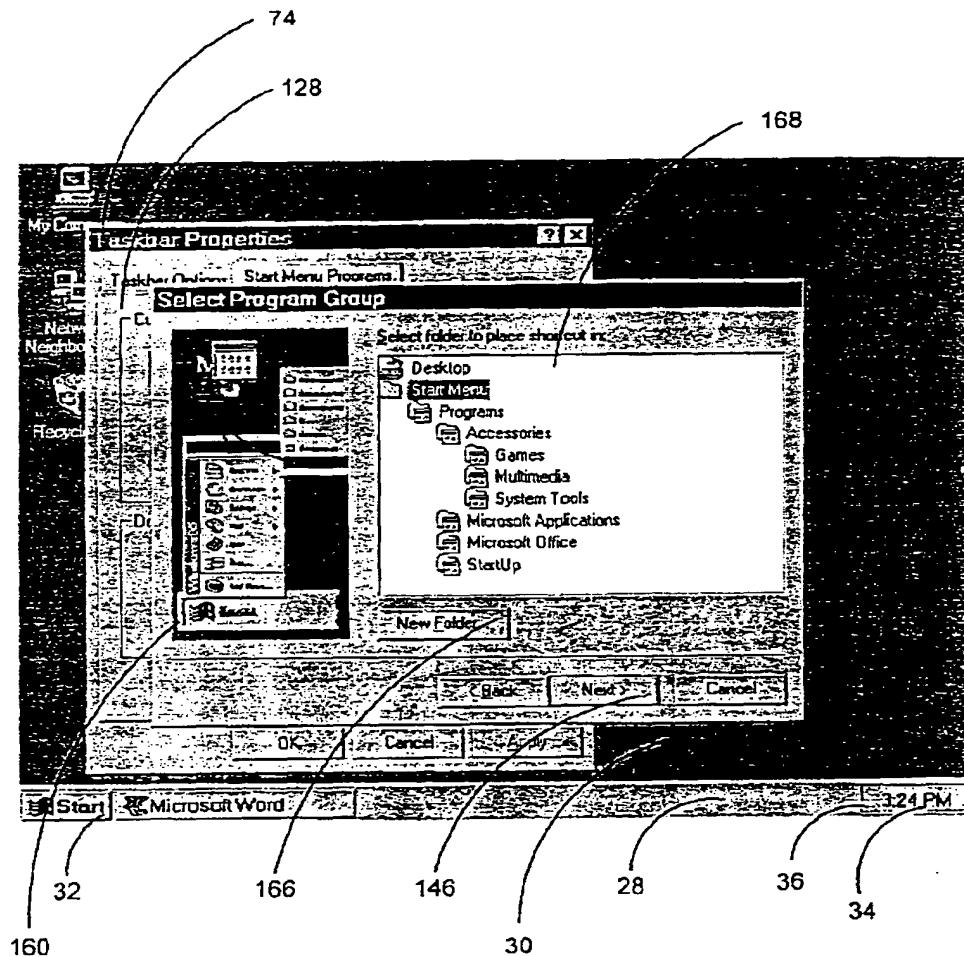


FIG. 24

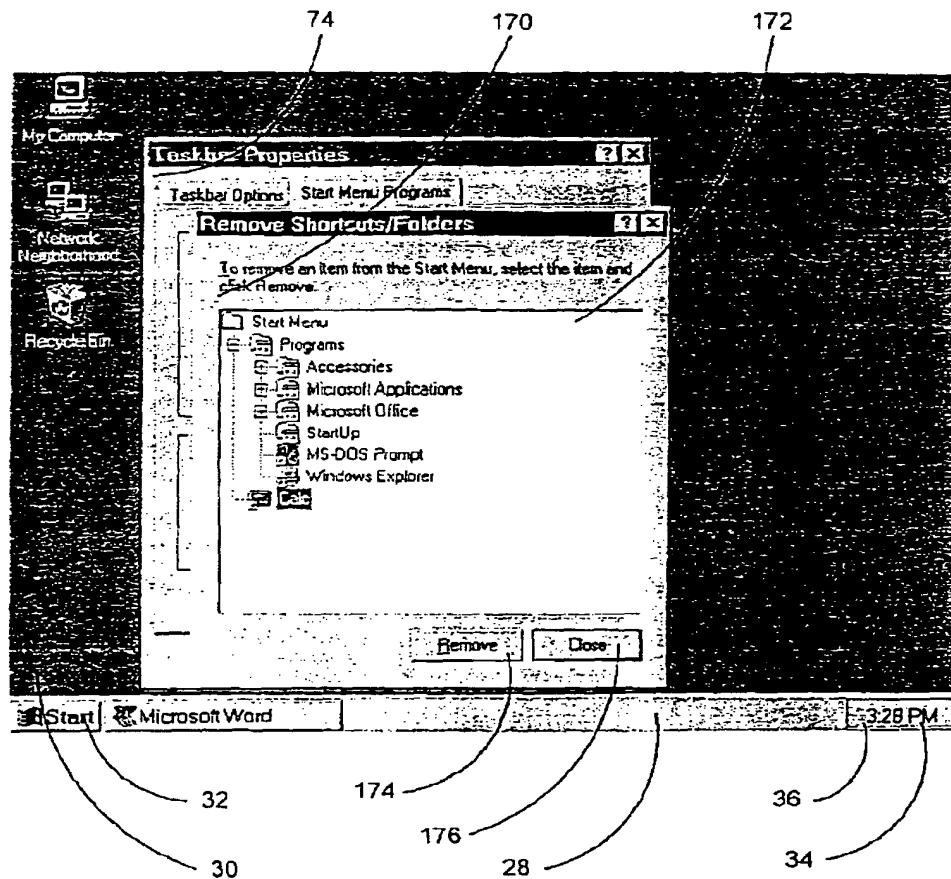


FIG. 25

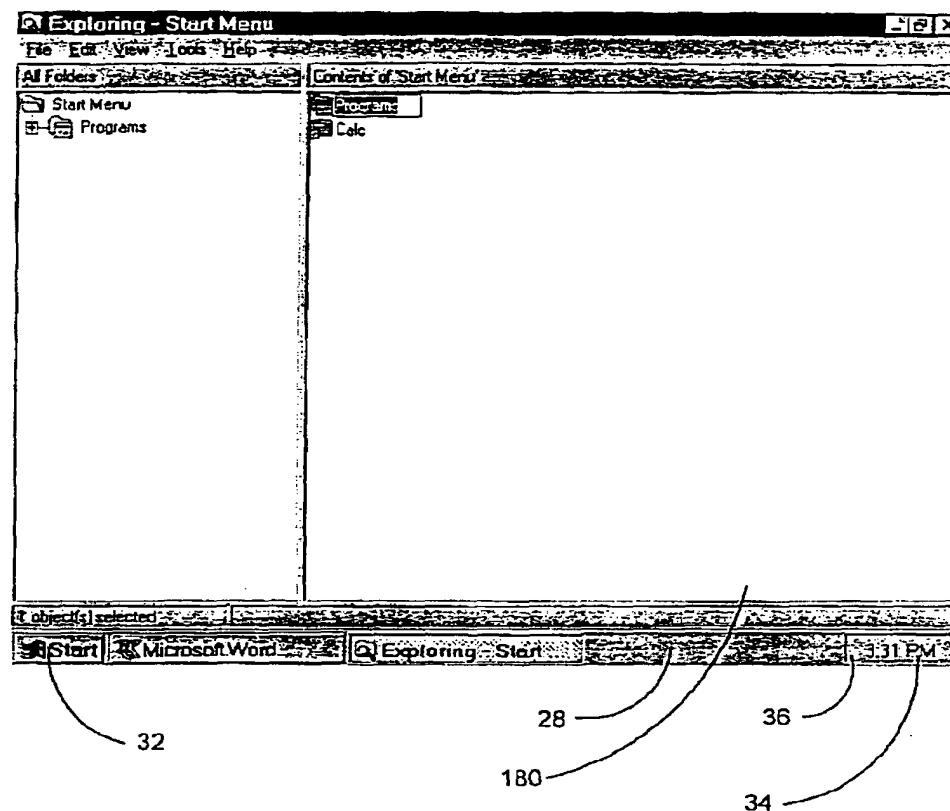


Fig. 26

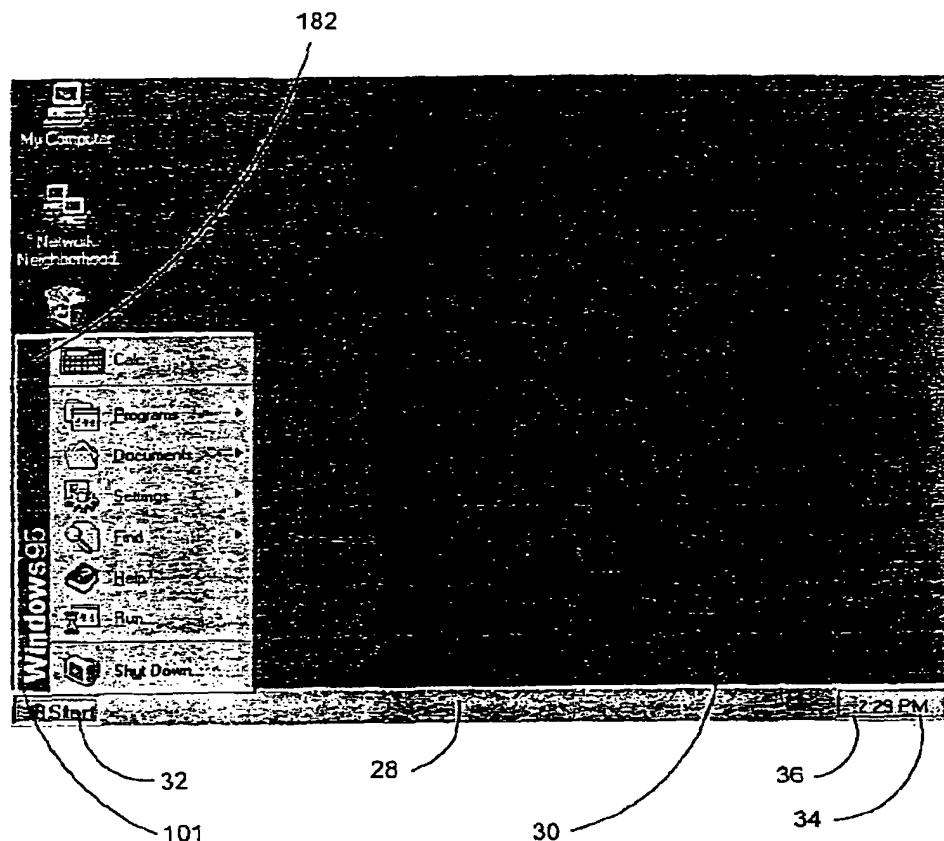


FIG. 27

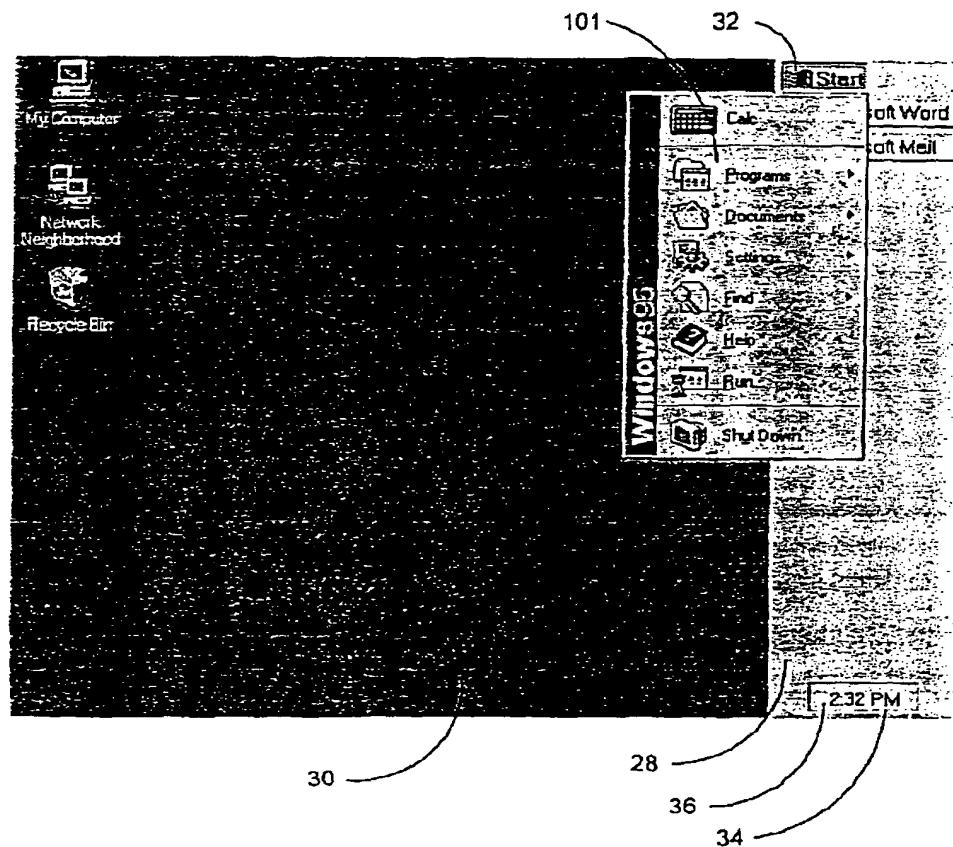


FIG. 28

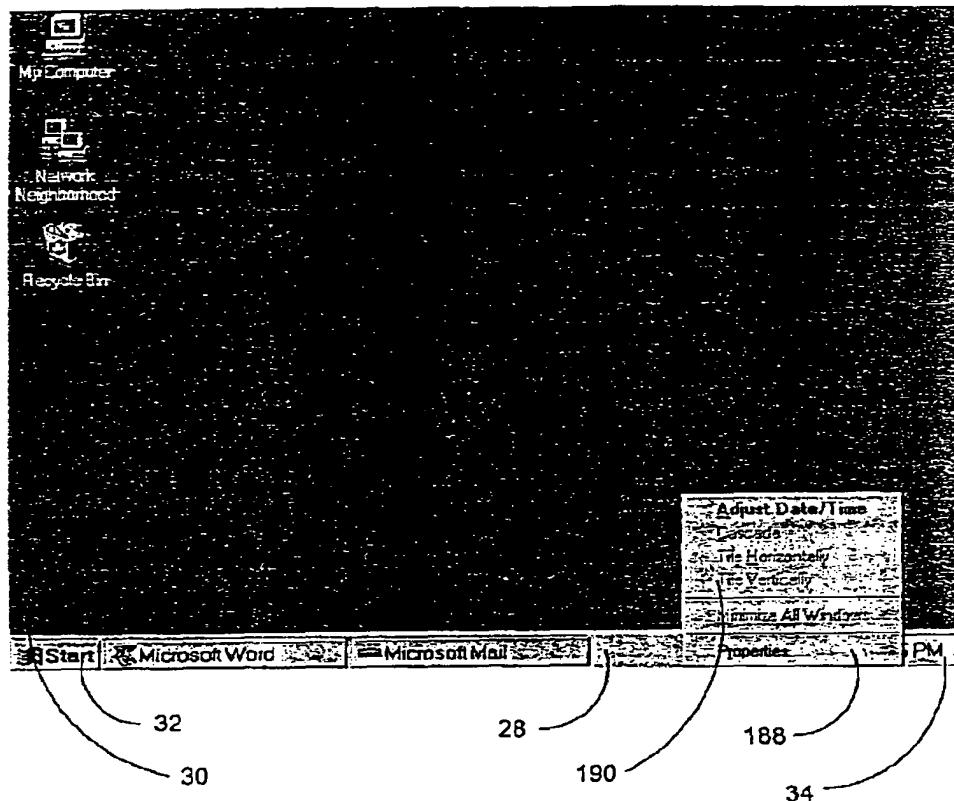


FIG. 29

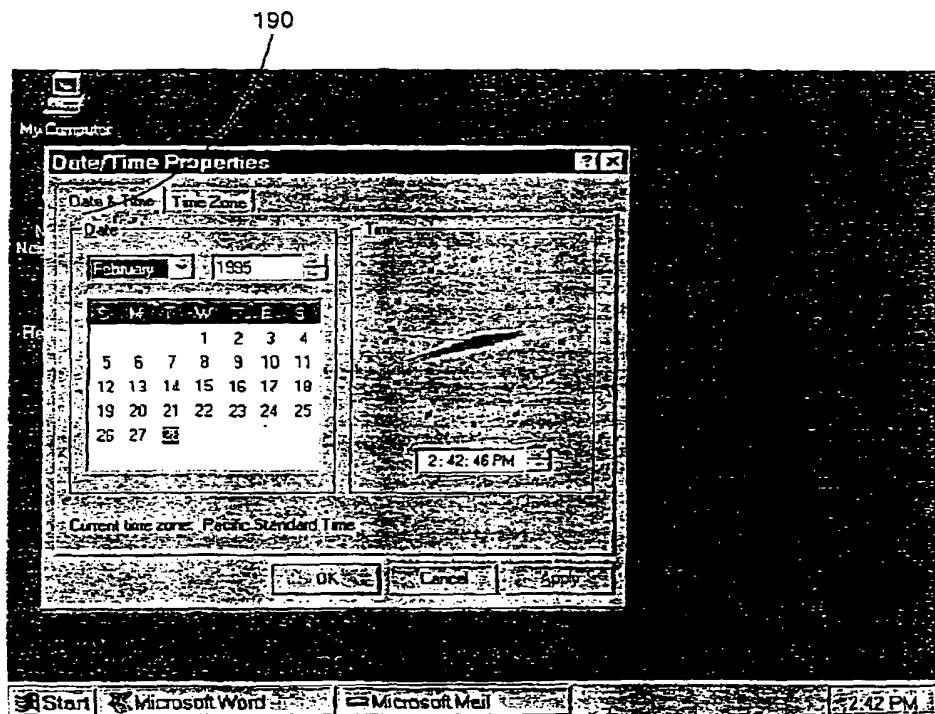


FIG. 30

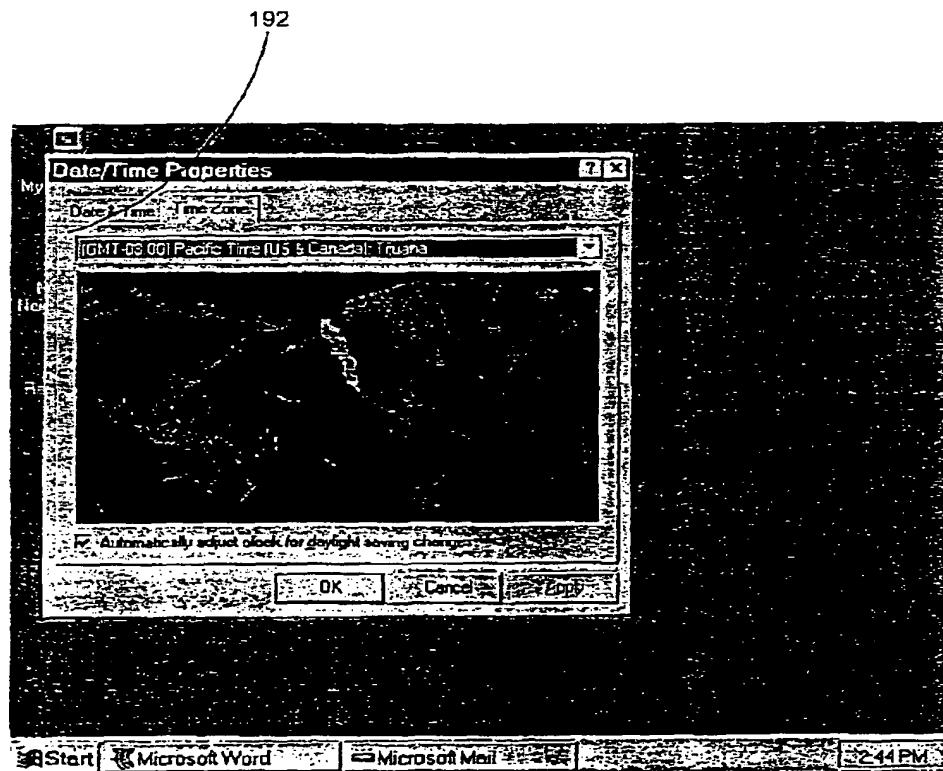


FIG. 31

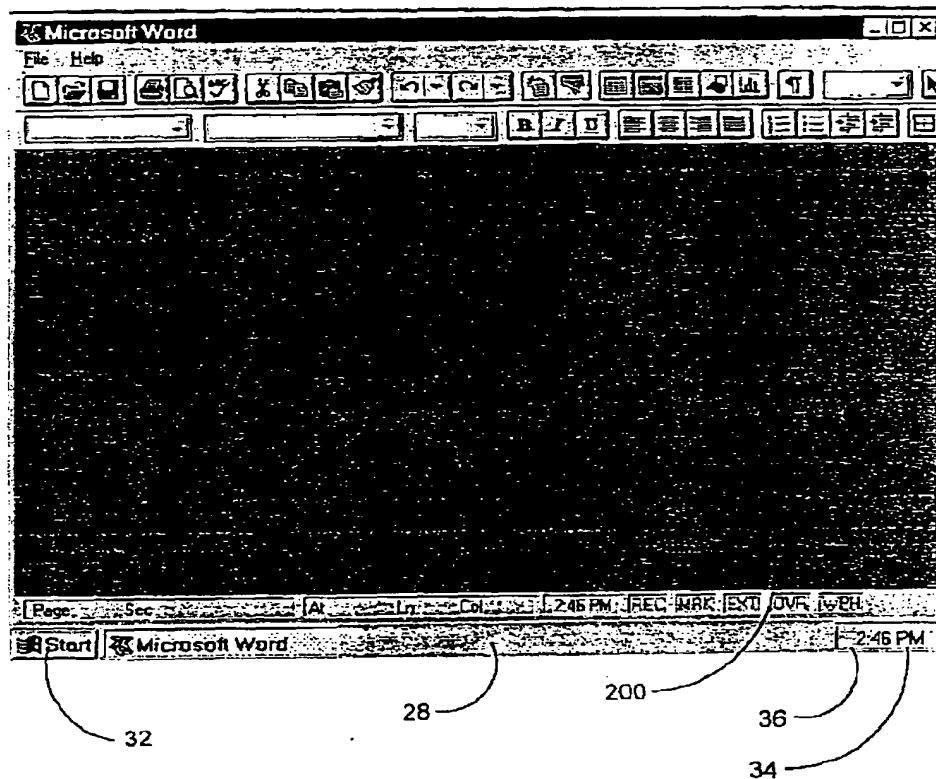


FIG. 32

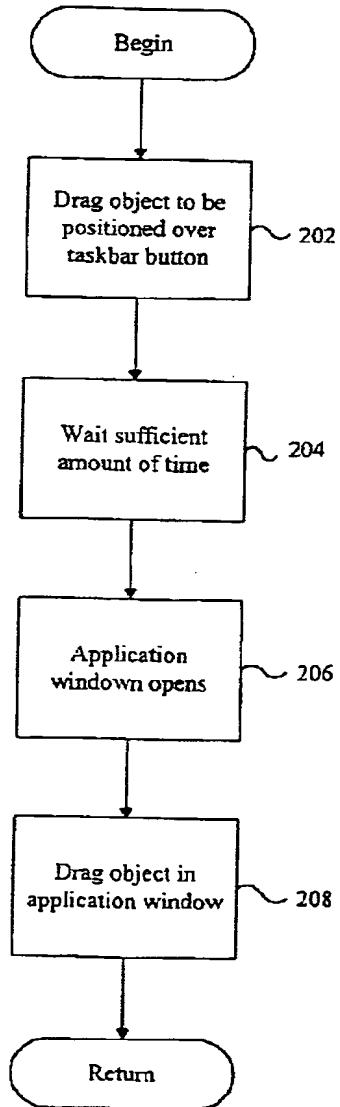


FIG. 33

1. Abstract

A taskbar is provided that supplies visual cues, such as buttons, for each application having an active window. These visual cues inform a user of the currently active windows and may be used to display the active windows. The taskbar may also include a start menu button that enables a user to gain access to a menu for launching programs, opening documents and controlling system settings. The taskbar is resizable and may be anchored to different locations on a video display. The taskbar may operate in a mode where it is not obscurable by other windows. The taskbar may also be displayed in a mode where it is not displayed when another window is in a non-minimized state.

2. Representative Drawing

Fig. 2